NAVAL FACILITIES ENGINEERING SERVICE CENTER
Port Hueneme, California 93043-4370

TECHNICAL REPORT
TR-2073-OCN

LOCAL/GLOBAL APPROACH TO NONLINEAR SIMULATION OF COMPLIANT MARINE STRUCTURES

by

Robert F. Zueck, Ph.D

June 1997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-018

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1997	Final 01 Oct 92-15 Dec 96

4. TITLE AND SUBTITLE

Local/Global Approach to Nonlinear Simulation of Compliant Marine Structures

5. FUNDING NUMBERS

6. AUTHOR(S)

R. F. Zueck, Ph.D

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSE(S)

Naval Facilities Engineering Service Center
1100 23rd Avenue
Port Hueneme, CA 93043-4370

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-2073-OCN

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESSES

Office of Naval Research
800 Quincy Street
Arlington, VA 22217-5660

Naval Facilities Engineering Service Center
1100 23rd Avenue
Port Hueneme, CA 93043-4370

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Also published as Ph.D. dissertation, University of California, Berkeley, 1996

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report presents reliable techniques for modeling extremely compliant structures. The research focuses on severe geometric nonlinearities associated with very large displacements and rotations. The solution requires two major modeling improvements: formulation of well-conditioned finite elements and development of specific control strategies for nonlinear step-by-step solution. Inherent in the physics of the structure, natural events condition the new finite elements. Associated event control directs the numerical solution to adhere closely to the true nonlinear structural response path. The numerical strategies are a simple extension of the trapezoidal rule for time integration and Newton iteration for nonlinear step-by-step solution. The result is extremely fast, efficient, and stable nonlinear structural simulation. A high level of computational robustness is essential for development of fully nonlinear substructured models. A local/global approach allows each substructure to have its own specialized local submodel and its own associated local solution strategy. A global model then integrates all the super-element representations of each diverse submodel. The local/global framework allows the nonlinear solution strategies to efficiently concentrate computational power where and when needed among the submodels. Code development and test problems focus primarily on compliant marine structures, where the need for robust, highly nonlinear simulation is so great.

14. SUBJECT TERMS

Geometric nonlinearities, nonlinear structures, nonlinear structural simulation, local/global approach, compliant marine structures

15. NUMBER OF PAGES

129

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

EXECUTIVE SUMMARY

This report focuses on reliable modeling techniques for extremely compliant structures. In particular, we implement the modeling techniques in a computer program called MBDSIM (Multi-body Dynamics Simulator) for use in simulating the system response of a ship towing a plow along the seafloor. The research pays particular attention to the severe geometric nonlinearities associated with very large displacements and rotations of deformable structures. The solution requires two major modeling improvements: formulation of well-conditioned finite elements and development of specific control strategies for nonlinear step-by-step solution. Inherent in the physics of the structure, natural events condition the new finite elements. Associated event control directs the numerical solution to adhere closely to the true nonlinear structural response path. The numerical strategies are a simple extension of the trapezoidal rule for time integration and Newton iteration for nonlinear step-by-step solution. The result is extremely fast, efficient, and stable nonlinear structural simulation.

A high level of computational robustness is essential for development of fully nonlinear substructured models. A local/global approach allows each substructure to have its own specialized local submodel and its own associated local solution strategy. A global model then integrates all the super-element representations of each diverse submodel. The local/global framework allows the nonlinear solution strategies to efficiently concentrate computational power where and when needed among the submodels.

Code development and test problems focus primarily on compliant marine structures, where the need for robust, highly nonlinear simulation is so great. However, the research results apply to many other types of structures.

This report is also published as a dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Engineering-Civil Engineering in the Graduate Division of the University of California, Berkeley (Zueck, 1996). The committee in charge of the Ph.D. dissertation included the following faculty at Berkeley:

- Professor Graham H. Powell, Chair
- Professor Gregory P. Fenves
- Professor Beresford N. Parlett

This research was funded in part by the Office of Naval Research and the Naval Facilities Engineering Service Center.

NOMENCLATURE FOR MATHEMATICAL SYMBOLS

Specific lists of mathematical symbols appear throughout this report. Arrays are constants or variables of any dimension, including the following one-, two-, and three-dimensional arrays: scalar, vector, and matrix. The following key clarifies the chosen nomenclature for all mathematical symbols found throughout this report.

$$\Delta x \qquad X_{bc}^{a} \qquad |\mathbf{x}| \qquad \mathbf{x}_{b}^{A} \qquad \mathbf{X}_{BC}^{-1} \qquad \mathbf{X}_{D}^{T} \qquad \mathfrak{I}(\dot{x}) \qquad \int_{\alpha}^{\beta} dx \qquad \sum_{\alpha}^{\beta} x'$$

The case and boldness of the main symbol have the following meaning:

$x \Rightarrow$ scalar referencing local, primary, or motion - based values

$X \Rightarrow$ scalar referencing global, secondary, or force - based values

$\mathbf{x} \Rightarrow$ array referencing local, primary, or motion - based values

$\mathbf{X} \Rightarrow$ array referencing global, secondary, or force - based values

Embellishments of the main symbol have the following meaning:

$\bullet \equiv$ nonspecific array ($x, X, \mathbf{x}$ or $\mathbf{X}$)

$\Delta\bullet \Rightarrow$ increment in time or space

$d\bullet \Rightarrow$ differentiation in time or space

$\dot{\bullet} \Rightarrow$ differentiation in time

$\bullet' \Rightarrow$ subset or subarray

$\bullet^{a} \Rightarrow$ (lower case superscript) raise to the power of a

$\bullet^{A} \Rightarrow$ (upper case superscript) special super - meaning

$\bullet^{-1} \Rightarrow$ (superscript -1) inverse

$\bullet^{T} \Rightarrow$ (superscript T) transpose

$|\bullet| \Rightarrow$ array norm (e.g., vector length)

$\bullet_{b} \Rightarrow$ (lower case subscript) term or component with index b

$\bullet_{bc} \Rightarrow$ (lower case subscripts) term of array at indices b,c

$\bullet_{B} \Rightarrow$ (upper case subscript) special sub - meaning

$\bullet_{BC} \Rightarrow$ (upper case subscripts) subarray with indices B,C

$\mathfrak{I}(\bullet) \Rightarrow$ nonspecific nonlinear function of $\bullet$

$\int_{\alpha}^{\beta} \bullet \Rightarrow$ integration from α to β

$\sum_{\alpha}^{\beta} \bullet \Rightarrow$ summation from α to β

ACKNOWLEDGMENTS

I wish to thank Dr. Graham Powell for his visionary leadership, research ideas, and philosophical insight. In addition, I wish to thank Dr. Beresford Parlett and Dr. Gregory Fenves for serving as distinguished members of this dissertation committee. I also wish to thank Dr. Robert Taylor for his personal encouragement, Dr. Randolph Paulling, Jr. for his enduring commitment, and Dr. Christopher Thewalt for his fine academic guidance. I am extremely fortunate to have worked with such inspiring and outstanding professors at the University of California at Berkeley.

I am grateful to Dr. Eugene Silva of the Office of Naval Research for arranging principal funding support and to Stephen Karnoski of the Naval Facilities Engineering Service Center (NFESC) for facilitating continued support. I acknowledge the pioneering efforts of Dr. Ronald Webster (Thiokol Corp.) and Dr. Ted Shugar (NFESC).

I dedicate this work in memory of Dr. William Nordell who initiated my post-graduate studies with a fellowship from the Naval Civil Engineering Laboratory. I also wish to thank David Shields and Norman Albertsen (NFESC) for their generous supervisory support. I am ever so grateful for the critical review that I have received from Paul Palo and Dr. Ray Chiou and all of my other fellow researchers at NFESC.

Finally, I would like to express my deep appreciation to Camille La Fredo, for her love and support during this modest endeavor in our life together.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1. INTRODUCTION

Rather than providing yet another numerical tool for incrementally improving the art of nonlinear structural analysis, we took a fresh look at the physics of the problem for a more revolutionary approach.

1.1 Need for Research

Structural analysis of compliant marine structures has become a very large and complex task. The underwater supporting structure for an acoustic measurement system (Zueck and Shields, 1995) provides an excellent example of this complexity.

As shown in Figure 1.1, the system consists of transmit, receive, and N-arrays for measuring the acoustic signature of a target submarine model. Fabricated mostly of aramid synthetic fiber, structural cables hold over 100 acoustic and electrical components at mid-depth in the 1,200-feet-deep lake. Submerged buoys near the lake surface hold up the arrays. Clump anchors on the lake bottom hold down the arrays. A steel cable to a winch on shore hauls down a buoyant target model for testing.

The structural system must be rather rigid to hold all acoustic components to less than one foot of a desired static position and to less than one inch of relative dynamic motion. On the other hand, acoustic transparency requires a minimum volume of the structural material. In summary, the perfect structural design requires an optimal balance between structurally rigidity (many large cables) and acoustic transparency (few small cables). It was computationally impossible to simulate the entire cable array as one model with enough detail to represent all significant structural components. Instead, several smaller models with varying levels of complexity and varying levels of scale had to suffice.

Examples of other compliant marine structures that often require sophisticated structural analysis include the following:

- Single-point moorings -- Estimating the "chaotic" drifting motion of sensor buoys and vessels with single-point moorings is essential for proper design.
- Vessel moorings -- Determining motion of ships and platforms with multi-leg mooring systems is necessary for assessing survival in severe weather.
- Towed vessels and arrays -- Computing the time-varying position and tension of towed underwater vessels and arrays are crucial to proper operation.
- Cable laying -- On-board, real-time simulation of how a cable lies down on an uneven seafloor is important for charting the course of the cable-lay vessel.

With increased need for complex analysis of compliant marine structures, nonlinear structural analysis has reached a computational barrier. Similar computational barriers exist with many other types of nonlinear structural analysis. This barrier characterizes itself with the following limitations on good structural analysis practice:

- Gross modeling simplification causes inaccurate analysis.
- Computational inefficiency causes expensive analysis.
- Numerical instability causes failed analysis.

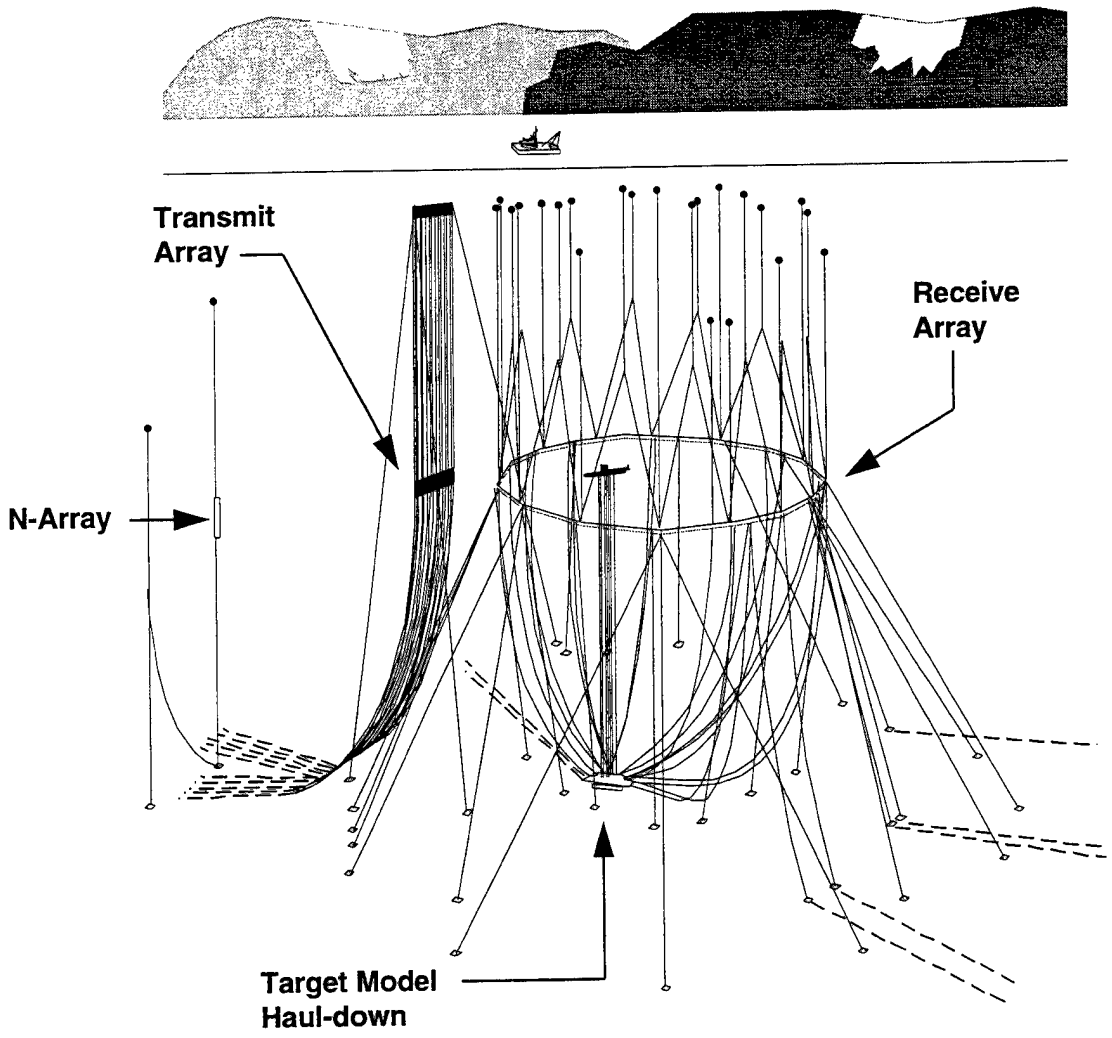


Figure 1.1 Complex compliant marine structure.

1.2 Research Objective

Our objective is to break the computational barrier that presently exists for analyzing the response of compliant structures. We abandon the present generic computational framework in favor of a more "physically structured" one. We reformulate the structural analysis problem so that it "naturally" lends itself to full modeling of nonlinear behavior, reduced computational expense, and solution stability. Structures are designed to be physically effective, economical, and stable. Therefore, the numerical models that represent them should be equally effective, economical, and stable.

To represent compliant structures, traditional nonlinear structural analysis relies on one large system of nonlinear equations based on one computational domain. These equations are often sparse, ill-conditioned, and poorly posed. The result is often a large number of trivial computations and divergence in the iterative nonlinear solution. For example, specific nonlinear

behavior of one small part of a large structure often dictates a very small time step for stable and accurate solution across the entire structural system.

Even though the mathematical community has conducted a significant amount of research on stable solution of sparse and ill-conditioned numerical systems, the computational barrier remains. We hypothesize that this barrier exists not for lack of excellent mathematical tools, but rather because the engineering models are not well posed to take proper advantage of the mathematical tools. The computational barrier exists, because structural analysts often formulate the physical model separately from physical considerations for how to solve the resulting nonlinear numerical system. We hypothesize that system nonlinearities do not permit us to separate the engineering model from its associated mathematical solution. Rather, the computational framework must integrally connect the two.

In summary, we seek to answer a specific question: *Can we formulate the nonlinear simulation of compliant marine structures in a computational framework that is inherently more robust?* We define each term to have the following meaning:

- *Nonlinear* means that each structural load can contribute in a highly disproportionate manner to the response.
- S*imulation* means predicting dynamic structural response in real-time.
- *Compliant* implies relative-motion, configuration-dependent loading.
- *Marine* denotes a diversity of structural elements with complex fluid loading.
- *Inherent* means natural physics and simple mathematics.
- *Robust* means efficient, fast, and stable step-by-step nonlinear solution.

1.3 Research Approach

We base the approach to our research on a novel divide and conquer philosophy and on a desire to move toward automated multi-domain structural modeling.

1.3.1 Divide and Conquer Philosophy. A traditional structural analysis model consists of a coupled system of discrete elements representing structural components at some representative model scale. Nodes represent the connections between elements. Solution of the model means computing nodal motions resulting from loads generalized to these nodes.

The choice of nodes and elements is rather arbitrary and relates mostly to the desired complexity of structural analysis. For example, we can choose from the following hierarchical scale of models for the offshore jacket shown on the far left of Figure 1.2:

(1) One cantilever element modeling the entire structure
(2) A series of frame elements that submodel the cantilever element
(3) An array of strut elements that submodel each frame element
(4) A series of beam-column elements that submodel each strut element
(5) An array of shell elements that submodel each beam-column element

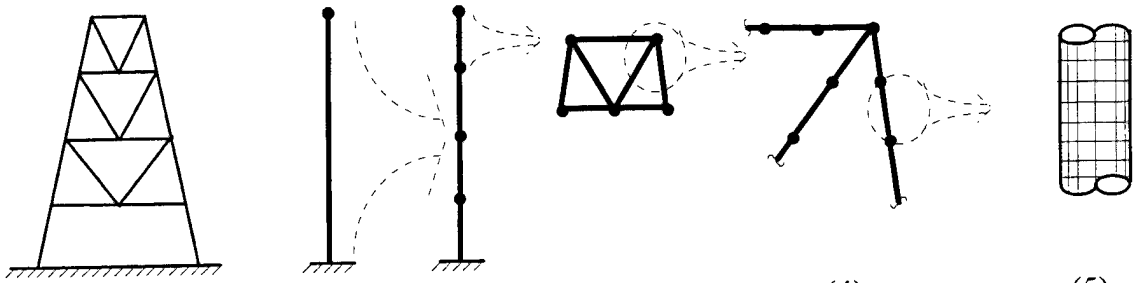


Figure 1.2 Hierarchy of model scales.

Model (1) of Figure 1.2, is useful for macro-scale global analysis of the entire structure. On the other hand, model (5) is useful for micro-scale local analysis of a specific substructure. Modeling the entire structure at the micro-scale is computationally prohibitive. For any given structure, it is necessary to balance technical complexity with computational expense when selecting appropriate modeling scale.

Structural analysts generally assign empirical relationships for structural behavior below the selected modeling scale. For example, an empirical relationship suffices for beam-column behavior of each strut element in model (3).

If the strut buckles in a way that is not consistent with this simple empirical relationship, then the finer scaled model (4) is necessary. Model (4) uses several beam-column elements for each strut to account for local buckling behavior. Stable solution of a model with these highly nonlinear buckling elements requires a special nonlinear solution strategy (Powell, 1989). Applying this special solution strategy to the entire model is computationally expensive and ill-advised.

A local/global approach to nonlinear structural simulation would be of great benefit for modeling the offshore jacket. The global model of the offshore jacket would be an array of strut super-elements as in model (3) of Figure 1.2. Each strut super-element would contain a local submodel as in model (4). The submodel would only be active for those struts that have or are approaching a buckling threshold. In a local/global framework, the solution shifts between local and global levels, which concentrates computational attention and power where and when needed. A computational framework that explicitly allows for this local/global approach to nonlinear structural simulation is preferable to the traditional one-domain approach.

1.3.2 Toward Automated Multi-Domain Structural Simulation. A multi-domain approach allows for optimal choice of modeling theory and solution strategy for each local substructure. Figure 1.3 shows how diverse submodels, each with a different kind of local specialization and mathematical formulation, can combine to form an integrated global model for the entire structure.

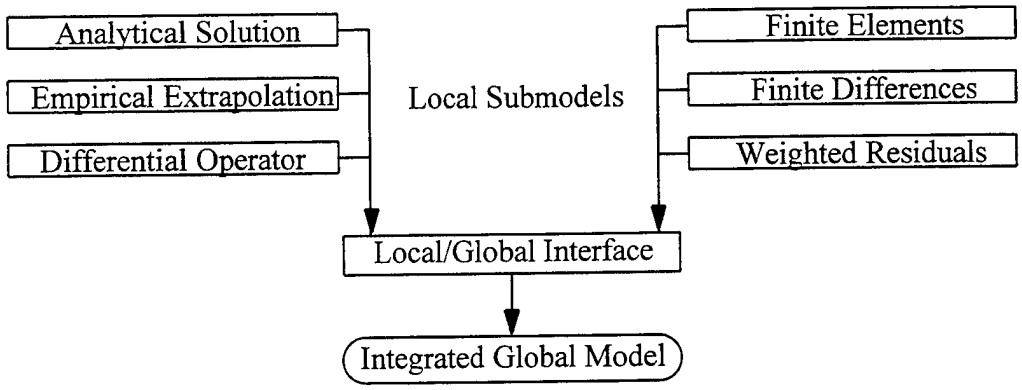


Figure 1.3 Multi-domain structural modeling.

Developing a full multi-domain computational framework applicable to a diversity of submodels, is not an easy task. Consequently, we focus on only two specific types of submodels, namely simple analytical and finite element submodels.

Along with obvious computational benefits, a local/global computational framework helps ultimately to automate certain portions of the engineering design and analysis process. In particular, a local/global approach provides for natural automation of the following local tasks for each substructure:

- Discretize the local domain into an optimal set of elements.
- Choose the best local loading patterns and effects.
- Represent complex local material properties.
- Select robust local solution strategies and parameters.
- Improve local analysis accuracy and reliability.

Automating these local modeling tasks will allow us to concentrate on more diverse investigations of the structural response. Rather than struggling with unreliable options for solution convergence, we can instead focus on the physical significance of the results.

The modern trend in software development is object-oriented programming. An object is a "black-box" module that has clearly defined data and action interfaces (Fenves, 1990). Utilizing this software paradigm, each submodel is a natural object.

A local/global approach provides for easy object-oriented organization of the entire computer code. Using a local/global interface, software developers can develop very complex submodels with specialized solution strategies at the local level without worrying about how these objects integrate into the structural model at the global level.

1.3.3 Focus on Compliant Marine Structures. Considering the many types of nonlinear structures that would benefit from a local/global computational framework, we focus principally on compliant marine structures. Consisting of several kinds of substructures with obvious physical differences (Zueck, 1986), compliant marine structures naturally lend themselves to a local/global model. In addition, the connectivity between substructures is often

very minimal. For example, a cable substructure requires a simple displacement-compatibility at each end of the cable.

Depicted in Figure 1.4, the deep-water semisubmersible platform (Zueck, et al., 1992) is an example of a compliant marine structure that naturally lends itself to a local/global computational framework. There are three obvious types of substructures that form this compliant ocean platform: semisubmersible, mooring, and anchor. The physics governing each type of substructure is drastically different. The semisubmersible is rigid and sensitive to loads from waves and wind. Cables are highly flexible and sensitive to hydrodynamic loads from current. Anchors interact with the soil and drag under high loads. However, all substructures act together as one integrated structure.

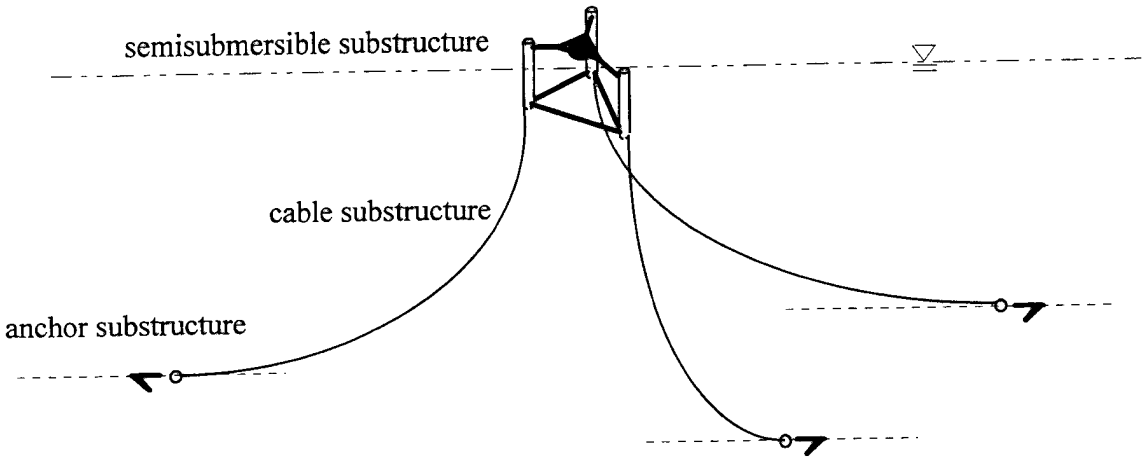


Figure 1.4 Semisubmersible platform structure.

The following submodels are suitable for representing each substructure:

- A rigid-body, large-rotation assembly of cylinders for the semisubmersible
- A flexible line of large-displacement finite elements for each cable
- An analytical expression of soil-structure interaction for each anchor

Figure 1.5 shows a simple local/global model of the deep-water semisubmersible platform. Global nodes define the connection between the semisubmersible, cable, and anchor submodels. Each cable submodel contains eight local nodes, connecting nine cable elements. For simplicity, the semisubmersible and anchor submodels contain no local nodes. In a more complex representation, these submodels may also contain local nodes.

Figure 1.4 depicts physical reality (i.e., the structure made up of substructures), while Figure 1.5 depicts the modeling approximation (i.e., the global model made up of submodels). Using the traditional one-domain computational framework, a large, sparse, and ill-conditioned matrix (depicted in Figure 1.6) results from the numerical model of the semisubmersible platform. In the matrix, the large "O" represents the global node of the semisubmersible submodel. Each smaller "o" represents a global node at an anchor submodel. The dots represent the local nodes for the cable submodels.

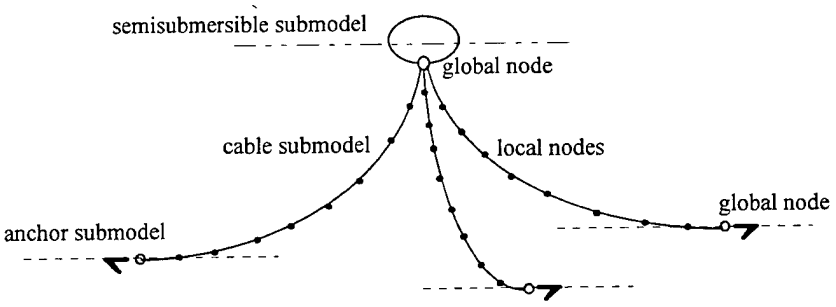


Figure 1.5 Semisubmersible platform model.

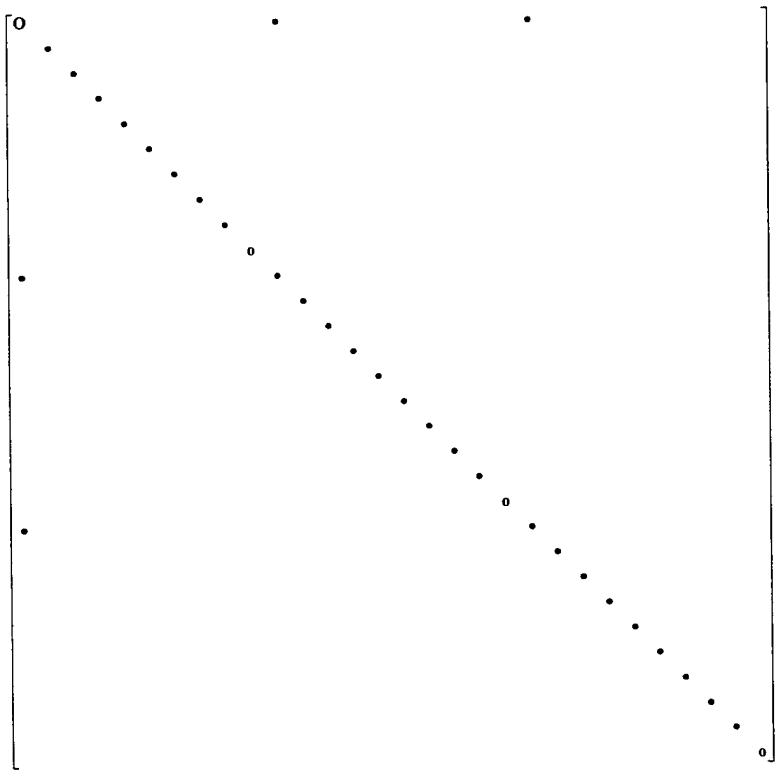


Figure 1.6 System matrix for one-domain approach.

In comparison to the large stiffness at global nodes, the local nodes have motion components (transverse to the cable) with very little stiffness. This large disparity in stiffness makes the matrix ill-conditioned.

We ask an obvious question: Why compose such a large, sparse, and ill-conditioned matrix? The alternative is to recognize the discrete submodels and never build the full global system. Each submatrix on the left side of Figure 1.7 represents a submodel. From left to right,

they are the semisubmersible, the three mooring cables, and the three anchors. Each submodel has its own local solution and condensation to a super-element.

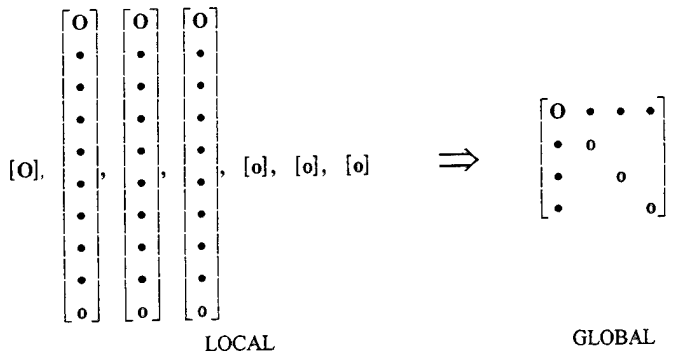


Figure 1.7 System submatrices for local/global approach.

At the global level, the seven super-elements assemble into the small dense matrix on the right of Figure 1.7. This small global matrix represents the coupling between all seven submodels. Instead of struggling with a large and sparse one-domain matrix, we work with smaller and denser local/global submatrices.

1.3.4 Specific Focus of Research. With obvious application to the broader field, we focus on a local/global computational framework and make the following modeling choices:

- Concentrate on deterministic simulation of compliant marine structures.
- Employ stiffness (displacement) method of structural analysis.
- Focus on cable substructures, with application to other types of substructures.
- Use finite or semi-analytical elements to represent substructures.
- Create stable elements that emphasize very large displacements and rotations.
- Build a framework for global integration of many submodels.
- Establish Newton-type solution strategies that allow full nonlinear modeling.
- Base iterative solution convergence on internal force equilibrium.
- Specialize the solution strategy for each specific submodel.
- Generalize the solution strategy for many types of global super-elements.

Technical literature is full of evolutionary improvements in the speed and efficiency of nonlinear computational methods, especially with regard to nonlinear material problems. However, this technical literature gives little attention to robustness and accuracy of nonlinear computational methods when applied to the geometrically nonlinear problem. Consequently, we focus particularly on robust numerical solution strategies for geometrically nonlinear structures (Zueck & Powell, 1995). The technical literature gives even less attention to varying boundary conditions (cable/seafloor interaction) and nonconservative excitation (hydrodynamics) in the presence of large displacements and rotations.

1.4 Summary of Report

The purpose of each remaining chapter of this report follows:

- Chapter 2 reviews traditional methods for nonlinear step-by-step simulation of compliant structures and the various limitations of traditional substructuring theories.
- Chapter 3 forms the critical ingredients of a new local/global computational framework for simulation of compliant marine structures. This includes a cable super-element based on a stable submodel of new finite cable elements.
- Chapter 4 develops a nonlinear solution strategy that is consistent with this local/global computational framework and exhibits revolutionary solution robustness.
- Chapter 5 describes new computer code for testing the local cable submodel.
- Chapter 6 provides specific test problems that demonstrate the efficiency, speed, and stability of the new cable submodel.
- Chapter 7 draws conclusions about the research, extends its application to other structures, and presents future implications.

2. REVIEW PRESENT TECHNOLOGY

To set the stage for a local/global computational approach to nonlinear structural analysis of compliant marine structures, we shall briefly review some of the basic principles of basic nonlinear structural analysis. Section 2.1 reviews general nonlinear structural analysis, while Section 2.2 concentrates specifically on marine aspects of nonlinear structural analysis. Section 2.3 presents some traditional concepts for substructured analysis, which form a limited basis for local/global simulation.

2.1 General Nonlinear Structural Analysis

Of the vast choice of structural analysis methods, we will concentrate on incremental nonlinear finite element methods. Unlike closed-form analytical methods, this temporal and spatial approximation permits full consideration of the nonlinear complexity of the physical problem.

2.1.1 Discrete Finite Element Formulation. The principle of virtual work or alternatively the principle of minimum potential energy, applied to a continuous structure leads to a mathematical set of initial value partial differential equations. These continuous equations have time as the independent variable and displacement as the principal unknown. To solve these differential equations and to satisfy associated boundary conditions, we choose to discretize the continuous structure in both time and space. This discretization process reduces the partial differential equation into a set of algebraic equations solvable by numerical computers.

2.1.1.1 Basic Dynamic Equations of Motion. The most popular method to discretize a continuous structure is the displacement-based finite element method. This method discretizes the spatial continuum into finite elements. Representing connections between

elements, each node has prescribed degrees of freedom representing the behavior of the elements connected to it.

In formulating elements for the finite element method, we utilize the three well-established principles of structural theory to relate discrete elemental behavior to overall structural behavior:

- Kinematic continuity: A compatibility relationship between element deformation and structural displacement
- Action/deformation: A constitutive relationship between element action and deformation
- Force equilibrium: A Newtonian relationship between nodal forces and element action

The element actions and deformations must be energy conjugate pairs of stress and strain or their resultants. The nodal displacements and forces must be physically consistent with the element actions and deformations. Element shape functions assure geometric compatibility.

By assembling all discretized element behavior according to the nodal connectivity, we get a single linear system of algebraic equations equating structural inertia, damping, and stiffness force to the externally applied force (load). Equation (2.1) shows this system of equations in matrix form (Clough and Penzien, 1975).

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{R} \qquad (2.1)$$

Where:

$\mathbf{M}$ = structural mass (generally symmetric & positive definite)

$\mathbf{C}$ = structural damping

$\mathbf{K}$ = structural stiffness

$\mathbf{R}$ = applied force (load)

$\ddot{\mathbf{r}}, \dot{\mathbf{r}}, \mathbf{r}$ = nodal acceleration, velocity, displacement

The primary generalized unknowns in equation (2.1) are the accelerations, velocities, and displacements at the nodes for each increment of time. Each set of motions corresponds to a displaced state of the structure as a whole and a deformed state of each finite element. The shape functions define the relationships between element deformations and structure displacements.

The load vector consists of contributions from both point loads applied at the nodes and point or distributed loads applied on the element. As a discrete system of algebraic equations, equation (2.1) satisfies the continuous differential equations in an average or "weak" sense.

2.1.1.2 Nonlinearity in The Equations of Motion. For compliant structures, all three basic structural mechanics principles are potentially nonlinear in the following manner:

- Kinematic continuity must include second order terms. For example, cables respond with a combination of large displacements and large rotations.

- Action/deformation relations are nonlinear. For example, synthetic ropes have nonlinear relationships between tension and extension.
- Force equilibrium is often dependent on the deformed configuration and loads may be configuration dependent. For example, beam-column buckling results from movement of the axial force.

Small displacement structural theory permits only the action-deformation relationships to be nonlinear. P-delta theory accounts approximately for second order effect in the force equilibrium calculation. Large displacement structural theory permits nonlinearity in both the kinematics transformations and in the force equilibrium calculation. A full nonlinear theory assumes nonlinear behavior in all three structural mechanics principles. We choose to use full nonlinear theory.

With large displacements come the possibility of displacement dependent constraints, such as element contact and gap formation. With displacement dependent constraints, we must check the constraint conditions and modify them as required for each step or substep of the solution. Although similar, steps represent major advances in load or time, while substeps represent minor corrections in load or displacement.

To allow for all possible nonlinearities, we reformulate the equations of motion (2.1) into an incremental form. In this incremental form, we linearize mass, damping, and stiffness matrices plus load vector for each progressive step or substep (Argyris and Mlejnek, 1991).

$$\mathbf{M}_n \Delta\ddot{\mathbf{r}}_n + \mathbf{C}_n \Delta\dot{\mathbf{r}}_n + \mathbf{K}_n \Delta\mathbf{r}_n \approx \Delta\mathbf{R}_n \tag{2.2}$$

Where:

n = step or substep number

$\mathbf{M}_n$ = structural mass (often constant)

$\mathbf{C}_n$ = structural damping

$\mathbf{K}_n$ = tangent structural stiffness

$\Delta\mathbf{R}_n$ = increment of force

$\Delta\ddot{\mathbf{r}}_n, \Delta\dot{\mathbf{r}}_n, \Delta\mathbf{r}_n$ = increment of nodal acceleration, velocity, displacement

In order to address large deformations in equation (2.2), it is popular to use a Lagrangian formulation. This formulation fixes its reference frame in space and allows material particles to move through it. This is in contrast to an Eulerian formulation that fixes its reference frame to the material particles. We generally use a Green and an Almansi strain tensor for the Lagrangian and Eulerian formulations, respectively, to describe material deformation (Belytschko, 1983).

For large geometric effects, it is popular to use an updated Lagrangian formulation, whereby all kinematic variables measure the deformed state relative to the previous state. This is in contrast to a total Lagrangian formulation, where all kinematic variables measure the deformed state relative to the initial undeformed state.

Crisfield argues that a simple rotated engineering strain formulation best suits compliant structures with large nonlinear geometric effects (Crisfield, 1991). However, if the formulation

is consistent in its inclusion of nonlinear effects and comprehensive in their implementation, all formulations theoretically yield identical results (Bathe, 1982).

2.1.2 Static Nonlinear Solution. By assuming zero acceleration and velocity, equation (2.2) reduces to the following system of static incremental algebraic equations.

$$\mathbf{K}_n \Delta \mathbf{r}_n \approx \Delta \mathbf{R}_n \tag{2.3}$$

We solve equation (2.3) by rewriting it in residual form.

$$\mathbf{K}_n \Delta \mathbf{r}_n = \mathbf{R}_n^U \tag{2.4}$$

Where:

$\mathbf{R}_n^U \equiv \mathbf{R}_n^E - \mathbf{R}_{n-1}^I$ = unbalanced force

$\mathbf{R}_n^E$ = externally applied force (configuration - dependent load)

$\mathbf{R}_{n-1}^I$ = internal resisting force (at end of previous step or sub - step)

The vector of internal resisting force is the nodal assembly of individual resisting forces internal to each element.

The essential goal of nonlinear static solution is to determine corresponding increments of displacement and load for which all static forces are in acceptable equilibrium at the end of each step. The Euclidean norm (maximum absolute value) of the unbalanced force generally gives the best scalar measure of this force equilibrium.

Prior to beginning a nonlinear solution, we must first establish an initial structural state, i.e., static shape and force equilibrium. In the case of geometrically nonlinear structures that do not have sufficient initial stiffness to support their own weight, the initial reference state is often not obvious.

At each step of the simulation, there may be one or more substep solutions for reducing force unbalance in the system. The system of linear incremental algebraic equations may potentially change at each substep of the solution. When solved step-by-step, the linear incremental equations are effectively continuously nonlinear.

2.1.2.1 Basic Incremental/Iterative Method. Most nonlinear step-by-step solution methods involve two phases of substeps:

- Incremental phase
- Iterative phase

The incremental phase consists of a set of substeps that predict the load increment for the step. The iterative phase consists of a set of substeps that correct for desired force equilibrium in the step. The number of corrective substeps (iterations) depends on the tolerance desired for

force equilibrium. There are numerous numerical strategies available for both the incremental and iterative phases of each step.

To advance past unstable limit points, these strategies may control the load increment, the displacement increment, or an arc-length combination of both (Simons and Powell, 1982). Figure 2.1 depicts these three choices by showing how each strategy incrementally advances along a continuous response path. For graphic simplicity, we choose a simple nonlinear path with only one degree of freedom. The figure depicts this continuous path as a solid line. The question mark in the load control sub-figure indicates that load control fails at a load limit. We cannot advance another load increment beyond this limit. Thus, load control does not work well for relaxing structures, which get more flexible with load and displacement. On the other hand, displacement control fails at a displacement limit, as indicated by the question mark in the second sub-figure. We cannot advance another displacement increment beyond this limit. Displacement control does not work well for stiffening structures, which get stiffer with load and displacement. The same limits do not apply to arc-length control. However, arc-length exhibits other problem-dependent failures (Carrera, 1994). The general conclusion here is that no one nonlinear solution strategy suffices for all structural problems.

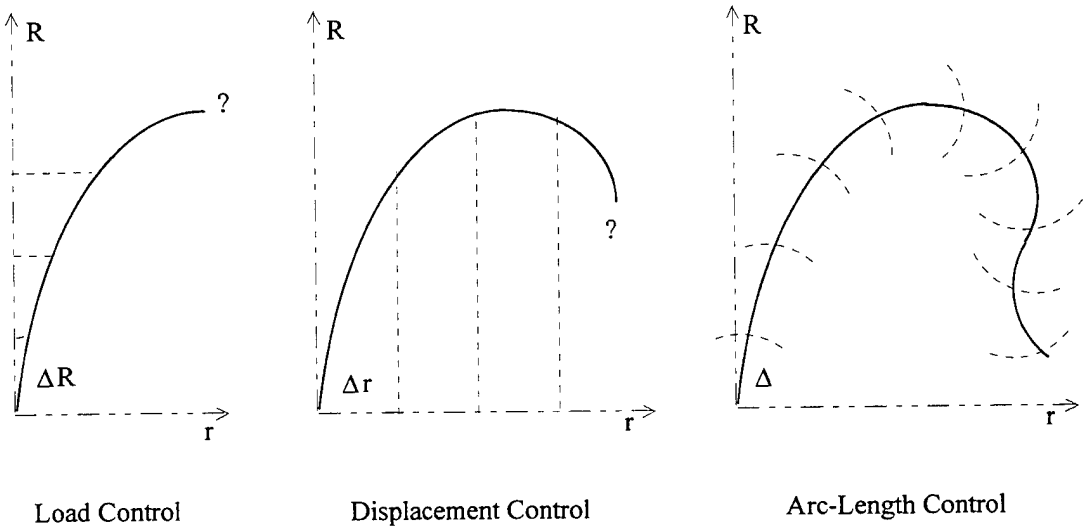


Figure 2.1 Incremental nonlinear solution control.

2.1.2.2 Newton-Raphson Procedure. Most incremental/iterative strategies for nonlinear solution are part of the Newton family of methods. The various members of the family differ in the way they update the stiffness. The following are three members of the Newton family:

- Modified Newton (initial or infrequently updated stiffness)
- Quasi-Newton methods (stiffness modified by a constraint)
- Newton-Raphson (updated stiffness)

Modified Newton methods hold the stiffness constant for a number of iterations. The motivation for using a modified Newton method is to reduce the number of expensive recalculations of the stiffness. Modified Newton methods generally sacrifice numerical stability for improved computational speed. Quasi-Newton methods modify the stiffness to accelerate solution convergence. A common modification involves under- or over-relaxing the actual stiffness. The most popular method for corrective iteration on the unbalanced force is the Newton-Raphson method. The Newton-Raphson method reformulates the stiffness at the beginning of each iteration.

Figure 2.2 depicts the Newton-Raphson procedure. For graphic simplicity in the figure, we show only one load step and only one degree of freedom. Also for graphic simplicity, we show only one substep for the incremental (predictive) phase and only three substeps for the iterative (corrective) phase. In the figure, the solution path is a light solid line and the true equilibrium path is a heavy solid line. In order to follow the true equilibrium path as closely as possible from point a to point b, we update the resulting stiffness at each substep.

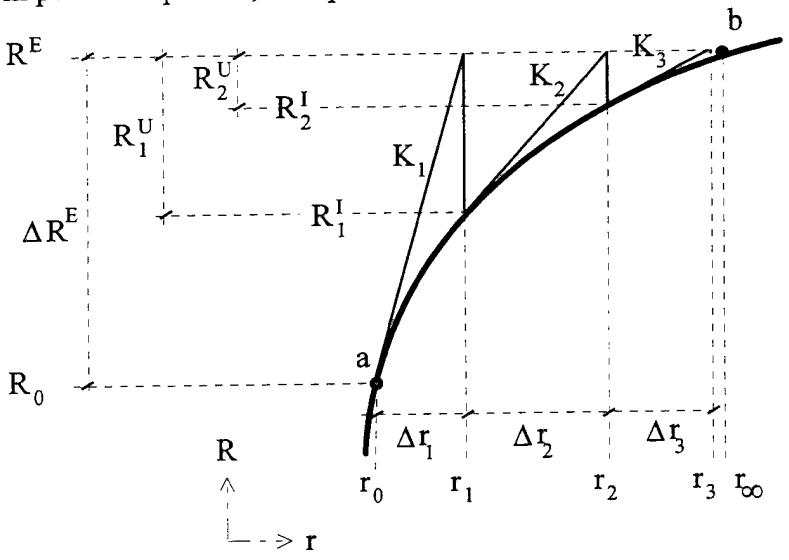


Figure 2.2 Newton-Raphson step.

Where:

$n = 1,2,3,...$ = substep number (0 = start of step)

$\mathbf{R}_n$ = total load

$\mathbf{r}_n$ = total displacement

$\mathbf{R}_n^U$ = unbalanced force

$\mathbf{R}_n^I$ = internal resisting force

$\mathbf{K}_n$ = tangent stiffness

$\Delta\mathbf{r}_n$ = increment of displacement

$\mathbf{R}^E$ = total load for step

$\mathbf{R}^E = \mathbf{R}_n \quad \text{as } n \to \infty$

$\Delta\mathbf{R}^E$ = load increment for step

Figure 2.3 shows the flow of information through the required computational modules for each successive load step. Since the full step may contain several substeps, the substep loops nest within the step loop. Since an incremental substep may contain several iterative substeps, the corrective loop nests within the predictive loop.

The list below describes each of the computational modules in Figure 2.3.

(0) Setup model:

Given an assumed initial state, set element actions and deformations.

$$\begin{aligned} &n = \infty \quad \text{(last substep in step)} \\ &\mathbf{r}_\infty = \text{given} \\ &\mathbf{R}_\infty = \text{given} \\ &\mathbf{S}, \mathbf{v} = \mathfrak{I}(\mathbf{r}_\infty) \\ &\mathbf{R}_\infty^U = \mathbf{0} \end{aligned} \tag{2.5}$$

(1) Increment load:

Increment load by a percentage of the total load.

$$\begin{aligned} &n \Leftarrow 0 \\ &\mathbf{r}_0 \Leftarrow \mathbf{r}_\infty \\ &\mathbf{R}_0 \Leftarrow \mathbf{R}_\infty \\ &\mathbf{R}_0^U \Leftarrow \mathbf{R}_\infty^U \\ &\Delta\mathbf{R}^E = \text{given} \\ &\mathbf{R}^E \Leftarrow \mathbf{R}_0 + \Delta\mathbf{R}^E \end{aligned} \tag{2.6}$$

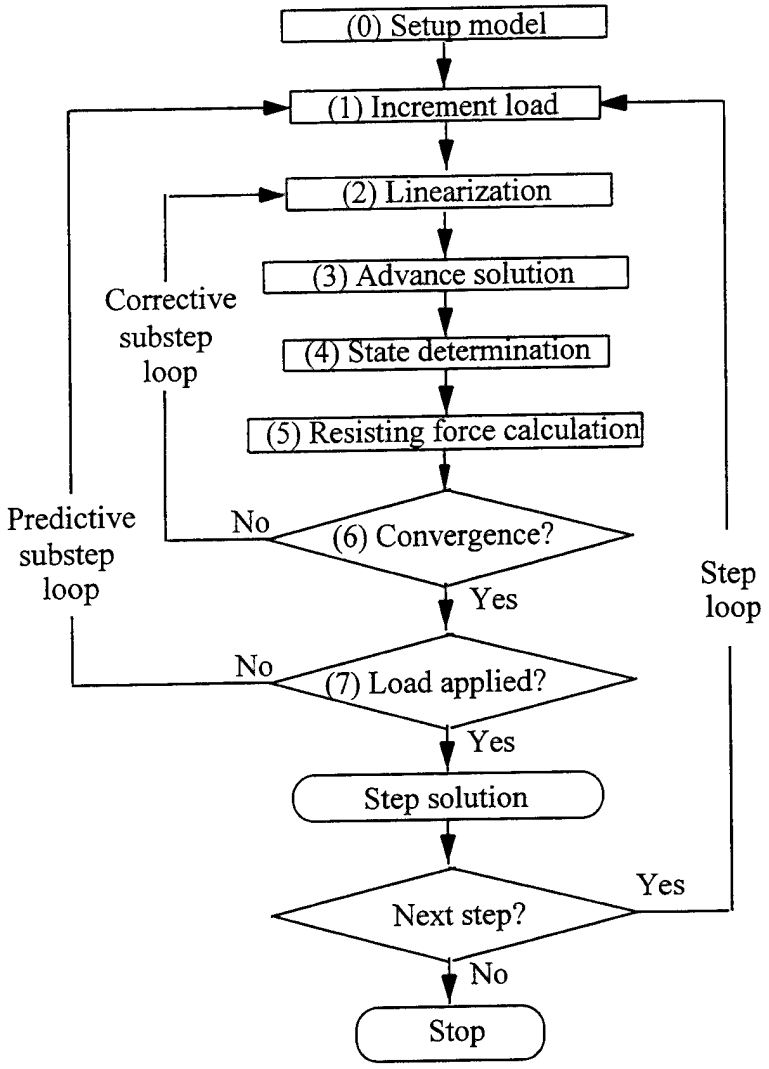


Figure 2.3 Traditional Newton-Raphson solution strategy.

(2) Linearization:

Compute tangent stiffness for all elements. Formulate tangent stiffness matrix.

$$n \Leftarrow n+1$$
$$\mathbf{K}_n = \Im(\mathbf{S}, \mathbf{v}, \ldots) \tag{2.7}$$

(3) Advance solution:

Formulate linearized force increment Assemble into a linearized matrix system.

$$\Delta \mathbf{R}_n = \Im\left(\mathbf{R}_{n-1}^{U}, \mathbf{R}^{E}, \mathbf{R}_0, \ldots\right)$$
$$\mathbf{K}_n \, \Delta \mathbf{r}_n = \Delta \mathbf{R}_n \tag{2.8a}$$

Select a displacement increment by a specialized control scheme or alternatively select a load increment based on all or a fraction of the total load, then calculate a displacement increment by solving linear matrix system. Update displacement.

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \Delta\mathbf{r}_n \tag{2.8b}$$

(4) State determination:

Calculate new actions and deformations.

$$\mathbf{S}, \mathbf{v} = \mathfrak{I}(\mathbf{r}_n, \ldots) \tag{2.9}$$

(5) Resisting force calculation:

Calculate nodal force for all elements and assemble into the internal resisting force.

$$\mathbf{R}_n^I = \mathfrak{I}(\mathbf{S}, \mathbf{v}, \ldots) \tag{2.10}$$

(6) Convergence?

Calculate the unbalanced force. If the unbalanced force has inadequate convergence, correct the solution by iterating through modules (2) through (6) using the unbalanced force to select a displacement or force correction.

$$\mathbf{R}_n^U = \mathbf{R}^E - \mathbf{R}_n^I \tag{2.11}$$

(7) Load Applied?

If system does not have full load applied, go to module (1) for new load increment.

2.1.2.3 Variations on Basic Incremental/Iterative Procedure. Size of the step and substeps can have a significant effect on the rate of solution convergence. Generally, the smaller the step or substep, the greater the chance of solution convergence. However, given a poorly posed starting state or a system that develops a severe ill-condition, even very small steps will not ensure convergence.

To improve the interval of convergence, there are numerous variations for the iterative phase (Simons and Powell, 1982), including the following:

- Variablè load step (iterating with a variable load step)
- Variable load direction (iterating with assurance of stability)
- State-dependent iteration (e.g., iterating only near the true state)

There are also strategies that alter the magnitude and/or direction of the displacement increment. For example, line search techniques select an alternate magnitude for the displacement increment in an attempt to minimize force unbalance.

Various modifications of dynamic and viscous relaxation can help deal with a non-equilibrium starting estimate or an ill-conditioned solution path (Underwood, 1983). Dynamic relaxation involves conditioning the static problem by temporarily defining pseudo mass and damping matrices. Viscous relaxation involves defining only the pseudo damping matrix. When the pseudo dynamic motion dies out, the static solution remains.

Although theoretically sound, viscous and dynamic relaxation does not work well in practice with geometrically nonlinear structures. This method results in long transients, excessive computations, and poor control of spurious oscillations (Webster, 1980). Automated dynamic relaxation, which involves a changing set of pseudo mass and damping matrices, shows better promise, but only given well-chosen automation properties (Shugar, 1988).

2.1.2.4 Solution of Linear Matrix System. To advance the solution, we must solve a linear matrix system of algebraic equations at each substep. Gaussian elimination or its variations (Friedberg, 1989) is the standard direct method for doing this. Gaussian elimination reduces a system of coupled algebraic equations to an equivalent set of uncoupled algebraic equations. In matrix notation, Gaussian elimination is the set of row and column operations for making the full matrix into a triangular, then a diagonal (uncoupled) matrix. As such, the number of computations relates directly to the number of equations.

We traditionally perform this elimination process in two steps. First, we eliminate all degrees of freedom with known displacements including displacements equal to zero. Then we perform normal Gaussian elimination to solve for unknown displacements.

Represented below in matrix form, the known degrees of freedom (subscript K) need not be part of the solution for the unknown degrees of freedom (subscript U).

$$\mathbf{K}_n \Delta \mathbf{r}_n = \begin{bmatrix} \mathbf{K}_{UU} & \mathbf{K}_{UK} \\ \mathbf{K}_{KU} & \mathbf{K}_{KK} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{r}_U \\ \Delta \mathbf{r}_K \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{R}_{UU} \\ \Delta \mathbf{R}_{KK} \end{bmatrix} = \Delta \mathbf{R}_n \tag{2.12}$$

By algebraic manipulation, we eliminate the known motions and leave a reduced matrix system.

$$\mathbf{K}_{UU} \Delta \mathbf{r}_U = \Delta \mathbf{R}_U \equiv \Delta \mathbf{R}_{UU} - \mathbf{K}_{UK} \Delta \mathbf{r}_K \tag{2.13}$$

By similar algebraic manipulation, we can get the unknown forces that correspond to the known degrees of freedom.

$$\Delta \mathbf{R}_{KK} = \mathbf{K}_{KU} \Delta \mathbf{r}_U + \mathbf{K}_{KK} \Delta \mathbf{r}_K \tag{2.14}$$

Most large structural systems are quite sparse; their stiffness matrices exhibit many zero terms. Matrix methods for solving a sparse system of algebraic equations that skip trivial computations can offer substantial computational savings. For well ordered, sparse structural systems, all non-zero terms may be on or near the matrix diagonal. "Banded" or "skyline" solution methods are available for these sparse systems (Row and Powell, 1978).

Iterative methods for indirectly solving a linear system of algebraic equations include the conjugate gradient or the Lanczos method (Nour-Omid, 1983). With proper pre-conditioning,

these methods can determine an adequate solution in a limited number of iterations, often making them more computationally efficient than direct methods.

Proper pre-conditioning means pre-multiplying the linear system with a special matrix, derived on the basis of improving the numerical condition of the system and on making the solution easier. If properly modeled, most compliant marine structures of interest do not possess a tremendous number of unknowns. As such, iterative methods do not offer much computational improvement over direct methods.

2.1.3 Dynamic Time Domain Solution. Dynamic solution is an obvious extension of static solution, just as the dynamic equations of motion (2.2) are an obvious extension of the static equations of motion (2.3). The only difference is that the dynamic solution requires a special treatment of the additional inertial and viscous forces. Two popular approaches exist for obtaining a dynamic solution, a frequency and time domain approach.

Requiring less computational effort, the frequency domain approach effectively linearizes the equations of motion over both time and space. With a frequency domain approach, we assume a linear set of harmonic vibrations for the dynamic solution (Clough and Penzien, 1975). An eigen-solution of the system of algebraic equations gives us a linear set of vibration frequencies (eigen-values) and associated mode shapes (eigen-vectors). For compliant marine structures, this simple steady vibration solution can be in error.

Recognizing the true temporal and spatial variability of compliant marine structures, the time domain approach preserves the unsteady nature of the dynamic solution. With a time domain approach, we solve equation (2.2) over time assuming a specific finite difference between time steps (Humar, 1990). Whereas the static solution has load steps, the dynamic solution has time steps.

We assume either an explicit or implicit finite difference between time steps (Newmark, 1959). Explicit methods use only historical information from past time steps. Implicit methods use assumed information for future time steps in addition to historical information from past time steps.

The greatest advantage of explicit difference methods is that they can result in element-by-element solutions with few computations at each time step. In other words, there is no need to solve a system of algebraic equations. The greatest disadvantage of explicit methods is that they require many small time steps relative to the structure's characteristic period for the temporal integration to remain stable over all time. Explicit methods are ideal for structural problems dominated by sudden wave propagation through a structure, for example, resulting from an explosive blast.

The greatest advantage of implicit methods is that they allow for a much larger time step relative to the structure's characteristic period than do explicit methods. The greatest disadvantage of implicit methods is that they require full system solutions and thus many computations at each time step. In contrast to explicit methods, implicit methods are useful for structural problems dominated by inertial effects such as compliant marine structures (Hughes and Belytschko, 1983).

Solution of the nonlinear dynamic problem starts by assuming a specific finite difference relationship between nodal acceleration, velocity, and displacement for any increment of time. Equation (2.15) gives this finite difference relationship for the Newmark family of one-step implicit methods (Bathe and Wilson, 1976).

$$\dot{\mathbf{r}}_n = \dot{\mathbf{r}}_{n-1} + \left[(1-\gamma)\ddot{\mathbf{r}}_{n-1} + \gamma\,\ddot{\mathbf{r}}_n\right]\Delta t$$
$$\mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{r}_{n-1}\Delta t + \left[\left(\tfrac{1}{2}-\beta\right)\ddot{\mathbf{r}}_{n-1} + \beta\,\ddot{\mathbf{r}}_n\right]\Delta t^2 \qquad (2.15)$$

Where:

n = 1,2,3,...= step or substep number

Δt = time step

β = Newmark weighting parameter for displacement

γ = Newmark weighting parameter for velocity

The Newmark family encompasses all one-step implicit time integration methods by appropriate choice of weighting parameters. The trapezoidal rule assumes constant average acceleration over the step by using the following weighting parameters:

$$\beta = 1/4$$
$$\gamma = 1/2 \qquad (2.16a)$$

Alternatively, the linear acceleration rule uses the following weighting parameters:

$$\beta = 1/6$$
$$\gamma = 1/2 \qquad (2.16b)$$

Solution stability is unconditional of time step size if the problem is linear and if the Newmark weighting parameters meet the following restriction:

$$1/2 \le \gamma \le 2\beta \qquad (2.17)$$

With finite element discretization, there is typically good accuracy in the lower frequencies of motion and progressively worse accuracy as the frequencies get higher. Newmark methods provide a good tradeoff between accuracy and numerical damping.

Parameter	Accuracy	Numerical Damping
$\gamma = 1/2$	$O^2(\Delta t)$	none
$\gamma \neq 1/2$	$O(\Delta t)$	yes

The following variations of the Newark method offer alternative tradeoffs of accuracy and numerical damping (Wood, 1987):

- Wilson-theta
- Hilber and Hughes
- Houbolt

Given a linear structural system, we can prove that any given one-step implicit method is numerically stable and accurate (Hughes, 1983) for a range of time step parameters. Unfortunately this proof does not extend to nonlinear structural systems. Instead, we can show general stability and accuracy only by executing a careful set of numerical test problems for each type of nonlinear structure and each set of parameters.

By appropriate algebraic manipulation of equation (2.15), we can produce expressions for incremental velocity and acceleration in terms of the increment of displacement. Then by substituting these expressions into equation (2.2) and rewriting in residual form, we get the following set of dynamic algebraic equations for incremental representation of the structural system.

$$\mathbf{K}_n^* \Delta \mathbf{r}_n = \mathbf{R}_n^U + \mathbf{R}_n^D \tag{2.18}$$

Where:

$\mathbf{K}_n^* \equiv a_1 \mathbf{M}_n + b_1 \mathbf{C}_n + c_1 \mathbf{K}_n$ = Jacobian (effective tangent stiffness)

$\mathbf{R}_n^U \equiv \mathbf{R}_n^E - \mathbf{R}_{n-1}^I$ = unbalanced force

$\mathbf{R}_n^E$ = externally applied force (configuration - dependent load)

$\mathbf{R}_{n-1}^I$ = internal resisting force (at end of previous step or sub - step)

$\mathbf{R}_n^D \equiv \left(a_2 \dot{\mathbf{r}}_{n-1} + a_3 \ddot{\mathbf{r}}_{n-1}\right)\mathbf{M}_{n-1} + \left(b_2 \dot{\mathbf{r}}_{n-1} + b_3 \ddot{\mathbf{r}}_{n-1}\right)\mathbf{C}_{n-1} + \left(c_2 \Delta \mathbf{r}_{n-1}\right)\mathbf{K}_{n-1}$

= constant force of time integration

The integration parameters for the Newmark family of methods are as follows:

$$\begin{aligned} a_1 &= \frac{1}{\beta \Delta t^2} & b_1 &= \frac{\gamma}{\beta \Delta t} & c_1 &= 1 \\ a_2 &= a_1 \Delta t & b_2 &= b_1 \Delta t & c_2 &= 0 \\ a_3 &= \frac{1}{2\beta} & b_3 &= \frac{\gamma \Delta t}{2\beta} - \Delta t \end{aligned} \tag{2.19}$$

Similar to nonlinear static solution, the essential goal of nonlinear dynamic solution is to determine corresponding increments of displacement and load for which all forces are in acceptable equilibrium. However in dynamic solution, the given time step encompasses the substep increments of displacement and load. In other words, the total load increment over a solution step must be the change in load over the time step.

Similar to nonlinear static solution, we must establish an initial structural state, i.e., static shape and force equilibrium. For reasons of stability, we generally choose the final static solution as the initial structural state for dynamic solution. We obtain the dynamic nonlinear solution over time in much the same way as the static nonlinear solution. However, for the dynamic step-by-step solution, we must include inertial and viscous forces and use the Jacobian as an effective stiffness matrix.

At each step of the simulation, there may be one or more substep solutions for reducing force unbalance in the system. The system of linear incremental algebraic equations may potentially change at each substep of the solution. When solved step-by-step, the linear incremental equations effectively give a continuous nonlinear response over time. Given the chosen finite difference relationship, we compute velocity and acceleration from corresponding increments of displacement.

2.1.4 Nonlinear Structural Analysis Codes. Of the many computer codes available for general nonlinear structural analysis, we will presently concentrate on one specific family of computer codes, named DRAIN.

2.1.4.1 Present Capability. Originally designed for nonlinear earthquake analysis, DRAIN includes three specific computer codes:

- DRAIN-BUILDING (Prakash and Powell, 1993)
- DRAIN-2DX (Prakash et al., 1992)
- DRAIN-3DX (Prakash et al., 1994)

Written in FORTRAN, each computer code has a base program (main routine) that controls all data management. The base program also controls the following types of analysis segments, applicable to compliant marine structures:

- Gravity: Linear static analysis for combined element loads and nodal loads
- Static: Nonlinear static analysis for nodal loads only
- Restore-to-static: Restore the structure to a static equilibrium state at the end of a dynamic analysis segment
- New dynamic force: Nonlinear dynamic analysis for force records
- Resume dynamic force: Continue dynamic analysis for another time segment

Procedures for adding elements to the base programs are logical and modular. DRAIN has the following library of elements:

- Truss bar (with limited use as a cable element)
- Plastic hinge beam-column
- Zero-length connection with translational and rotational hysteresis loops
- Inelastic gap element

None of these elements account for full large displacement and large rotation geometric effects. However, the beam-column element does account for P-delta type nonlinearity.

DRAIN manages the data between the base program and the elements using a set of labeled common blocks, stored in permanent and temporary computer files. Some common blocks transfer information between the base program and element subroutines as part of the well-defined and modular element interface.

Within any load segment, the base program performs a step-by-step nonlinear static simulation with event-to-event control (Porter and Powell, 1971). With this solution scheme, the

base program selects a substep, by determining when the next stiffness change (event) occurs and ending the substep at that event. The base program then modifies the structure stiffness and proceeds to the next substep.

The DRAIN-BUILDING code provides for a simple substructuring provision in terms of floors and inter-floors. The base program establishes a separate data structure for each floor and each inter-floor, thus preserving them as distinct substructured objects. In addition, the base program partitions the full structure stiffness matrix into related floor and inter-floor hyper-matrices.

The DRAIN-2DX and DRAIN-3DX codes store the stiffness matrix in compacted column or skyline form. This involves omitting the terms above the first non-zero term in each column and storing all terms below, up to, and including the diagonal term. The DRAIN-BUILDING code uses a hyper-matrix variation of compacted column storage. DRAIN uses a modified Crout factorization (Row and Powell, 1978) to solve the system of algebraic equations. The DRAIN-BUILDING code has the option of using a hyper-matrix solution method (Fuchs, et al., 1972), but for static solutions only.

2.1.4.2 Deficiencies in Present Capability. Transient computer simulations can be unstable and computationally expensive. Most methods rely on an initial estimate of the stiffness of the structure. This infers that a stable initial and current state is definable.

By using the event-to-event nonlinear solution strategy, DRAIN seeks to prevent significant unbalanced force from developing. However, the strategy does not always achieve this goal. If elements have curvilinear behavior or if axial force changes rapidly, there may be significant force unbalance. There is currently no direct provision for iteration with the event-to-event solution scheme.

If there are element loads, DRAIN does not permit nonlinear behavior. The reason is that element loads can take any form. As such, DRAIN cannot generalize them in a consistent fashion. If an element load causes nonlinear behavior, DRAIN would have to recognize and account for this behavior at the element level.

DRAIN currently ignores all off-diagonal terms for mass and damping. Consistent mass becomes essential for complex substructured models, because the substructures can have substantial internal inertia effects. DRAIN also does not currently provide consideration for inertial force and viscous damping internal to the substructures in the current element interface. Nevertheless, DRAIN could be a viable base program for modeling substructures, if we extend the element interface to allow for super-elements made up of a submodel of elements.

2.2 Marine Structural Analysis

2.2.1 Structural Modeling. Compliant marine structures require a wide variety of physics theories, structural models, and numerical solution methods (Shugar and Armand, 1987). Cables are a principal substructure for connecting other marine structures and making them compliant. Therefore, we will concentrate our review on submodels for cable structures. There are obvious extensions to general nonlinear structures.

2.2.1.1 Cable Catenary Formulation. We can generate discrete catenary properties such as shape, tension, and tangential stiffness using numerical solutions of implicit

analytical formulae. Classical derivations based on different implicit parameters are available (Irvine, 1981). A simple cable catenary formulation restricts solution to an non-extensible uniform cable supported at its ends and loaded solely by a uniform weight as shown in Figure 2.4.

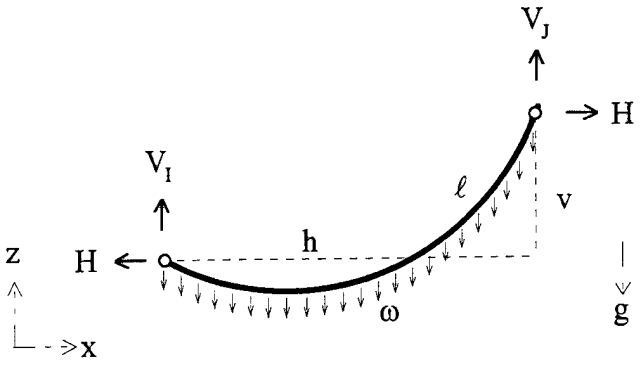


Figure 2.4 Basic catenary formulation.

Where:

h, v = horizontal and vertical dimensions of catenary
H, V = horizontal and vertical force at cable ends I and J
ℓ = cable length
ω = load per unit length
x, z = local coordinates
g = direction of field load

We choose a typical set of parametric equations (Peyrot & Goulois, 1979) for representing an inelastic cable catenary with simple uniform load.

$$\begin{aligned} \ell^2 &= v^2 + h^2 \frac{sinh^2(\lambda)}{\lambda^2} \\ V_I &= \frac{\omega}{2}\left(-v\frac{cosh(\lambda)}{sinh(\lambda)} + \ell\right) \\ V_J &= \omega\ell - V_I \end{aligned} \tag{2.20}$$

Where:

$$\lambda = \left|\frac{\omega h}{2H}\right|$$

Equation (2.20) involves six basic parameters, three of which we must choose as known. Given three parameters as known, we may need to solve for the other three parameters in an

iterative fashion. However, there is no guarantee of a solution (Levinson and Kane, 1993). The point to make here is that even the simplest of analytical solutions for a single catenary cable is mathematically complex.

To formulate and solve closed-form analytical solutions, it is often necessary to make extremely restrictive assumptions about the physics of the cable. In contrast, numerical solutions do not have such a limit and thus are generally superior for representing cables. In addition, we do not need to make any assumptions that limit the nonlinear complexity of the basic physics that govern the behavior of the cables.

2.2.1.2 Numerical Systems Approach. There are several physical characteristics of marine cable systems that can develop extremely nonlinear relationships in the structural analysis equations (Webster, 1979). We classify these nonlinearities into seven distinct categories:

- Geometric Nonlinearity: Cable systems may need to change shape dramatically to develop any substantial resistance to loads.
- Position-Dependent Loads: Loads depend on the orientation and position of each structural element.
- State-Dependent Loads: Fluid-induced loads are nonlinear with respect to velocity.
- Position-Dependent Constraints: Structural elements may interact with rigid or semi-rigid boundaries such as the seafloor.
- Nonlinear Materials: Constitutive relationships between stress, strain, and strain rate can be nonlinear.
- Physical Alteration of the Structure: The cable structure itself may change (e.g., adding new elements).
- External Structure Interaction: Ships, buoys, underwater vessels, and other relatively rigid bodies may interact in a nonlinear fashion with cable moorings.

Figure 2.5 shows a typical nonlinear response for the semisubmersible platform moored in deep-water subjected to static forces of wind, wave drift, and current loads (Zueck and Shields, 1984). The platform hull moves substantially off-station and the mooring legs take shapes very different from their dead-weight profiles.

Methods for numerical representation of nonlinear cable systems vary from lumped parameter methods to finite element methods (Leonard, 1988). The first successful application of an iterative procedure for nonlinear static analysis of cable systems was for the shape calculation of highly pre-tensioned two-dimensional cable roofs (ASCE, 1971). Some of the earliest analysis procedures were similar to the modern nonlinear displacement-based finite element method (Argyris and Scharpf, 1972).

There was obvious frustration with the stability of these finite element methods particularly when applied to three-dimensional cable networks with low pre-tension (Leonard and Recker, 1972). As such, alternative procedures, such as semi-analytical, catenary expressions (Peyrot, 1979), gained popularity.

These semi-analytical methods are efficient for analysis of electrical transmission lines (Peyrot, 1978) where we simply must determine catenary shapes under simple static gravity load.

However, these methods have limited application for marine cable structures (Peyrot, 1980) principally because of restrictions in applying full hydrodynamic loads.

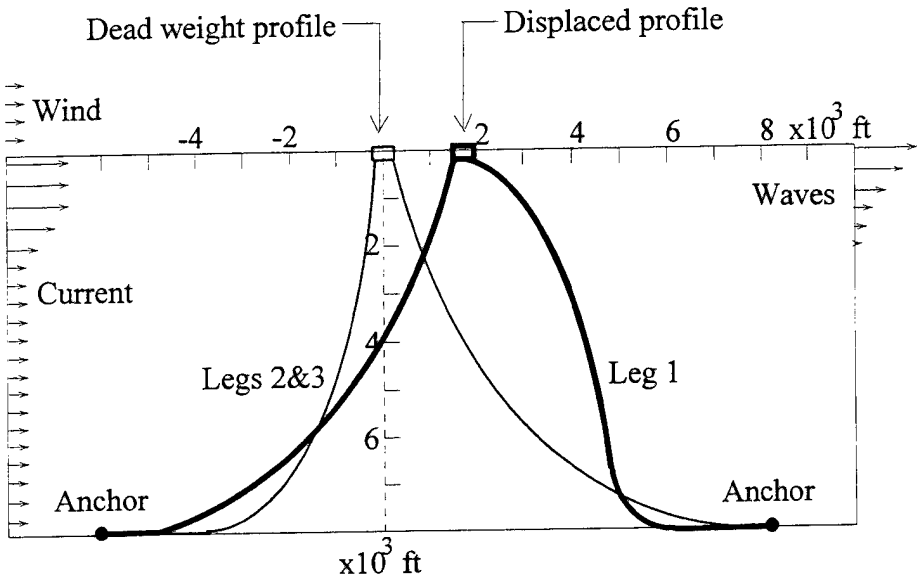


Figure 2.5 Static response of deep-water semisubmersible platform.

Numerical integration spatially along a cable offers a unique method for modeling cables (Chiou, 1989). Formulated using the full dynamic differential equation of a cable with full nonlinear boundary conditions, this cable model shows excellent potential for handling highly nonlinear spatial behavior. This includes spatially varying hydrodynamic load and selective seafloor contact.

The most general-purpose finite element model for dynamic analysis of marine cable systems uses the standard truss (bar) element formulated for large displacements and hydrodynamic loads (Webster, 1975).

2.2.2 Hydromechanics Modeling. The Navier-Stokes equation, which represents a balance of inertia, body, pressure, and viscous fluid force, governs general hydromechanics modeling. When coupled with structural equations, Navier-Stokes equations provide full fluid/structure interaction. Requiring often thousands of elements to discretize the fluid/structure continuum, this full interactive model is ill advised for even the simplest of marine structures.

Fortunately for most compliant marine structures, a one-way interaction suffices. As such, we recognize displacement of the structure, but ignore the displacement of the fluid. Even with this one-way interaction, fluid forces play both a loading and a resisting role. For fluid moving past a still structure, the fluid adds energy as a loading force on the structure. On the

other hand, for a structure moving through still fluid, the fluid removes energy as a resisting force on the structure.

The structure may comply to the fluid forces. In other words, the structure may displace appreciably and reduce or otherwise change the fluid forces acting on the structure. This relative motion between the moving fluid and the moving structure is a major nonlinearity in the hydrodynamic modeling.

In parallel with structural theory, we prefer to classify fluid forces in three categories, namely inertia, damping, and stiffness as shown in equation (2.21). Recognizing the relative motion between the fluid and the structure, we compute fluid forces by integrating over all the structural surfaces in contact with fluid.

$$\mathbf{R}^H = \mathbf{R}^M + \mathbf{R}^C + \mathbf{R}^K = \text{fluid force at element nodes} \qquad (2.21)$$

Where:

$$\mathbf{R}^M = \int_\Omega \mathfrak{I}(\ddot{\mathbf{r}}, \dot{\mathbf{u}}) = \text{hydrodynamic inertial force}$$

$$\mathbf{R}^C = \int_\Omega \mathfrak{I}(\dot{\mathbf{r}}, \mathbf{u}) = \text{hydrodynamic damping force}$$

$$\mathbf{R}^K = \int_\Omega \mathfrak{I}(\mathbf{r}, \grave{\mathbf{u}}) = \text{hydrostatic stiffness force}$$

$\ddot{\mathbf{r}}, \dot{\mathbf{r}}, \mathbf{r}$ = acceleration, velocity, and displacement of structure

$\dot{\mathbf{u}}, \mathbf{u}, \grave{\mathbf{u}}$ = acceleration, velocity, and displacement of fluid

Ω = all fluid / structure surfaces

2.2.2.1 Basic Hydrostatics. Hydrostatic force is the integration of static fluid pressure on all fluid/structure surfaces. For very slender members, we can idealize this distributed hydrostatic force as an equivalent point force at each element end. These point forces act in the vertical direction opposite gravity as shown in Figure 2.6.

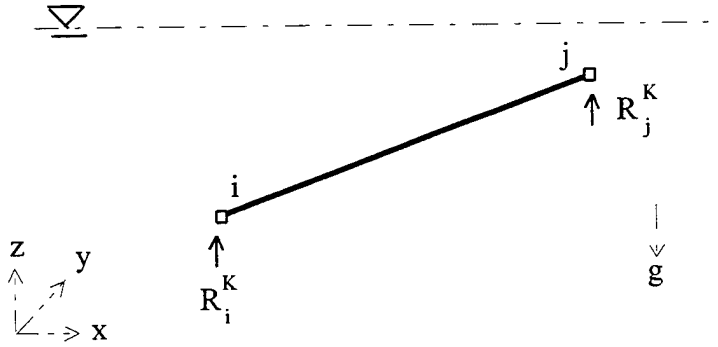


Figure 2.6 Simple hydrostatic/gravity force.

Given that the slender member is fully underwater, the hydrostatic/gravity forces on the element ends are equal and unchanging.

$$R_i^K = R_j^K = \frac{1}{2}(\gamma_H - \gamma) A \ell_0 \qquad (2.22)$$

Where:

γ_H = unit weight of fluid

γ = unit weight of member

A = cross-sectional area of member

ℓ_0 = nominal length of member

If the member pierces the fluid surface, then hydrostatic force becomes a stiffness-type force that is dependent on displacement. If the member contains fluid (e.g., oil in a pipeline), then the hydrostatic force takes on an even more complex form (PMB, 1989).

2.2.2.2 Basic Hydrodynamics. Because of the intractability of solving the highly nonlinear Navier-Stokes equations, there are "engineering" theories for representing hydrodynamics on marine structures. These theories are semi-empirical and use concepts such as added mass and fluid damping. Added mass relates to the conservative inertia force created by fluid acceleration around a structural member. Fluid damping relates to the nonconservative force created by fluid velocity around a structural member. Traditionally, we choose from two different "engineering" theories -- diffraction theory and viscous theory.

Diffraction theory assumes an ideal fluid possessing inviscid and irrotational flow. In this theory, we replace the normal governing partial differential equations in three unknowns (velocity components) by equations in one unknown (velocity potential). Although this velocity potential greatly simplifies the true physics of fluid flow, it provides for easy numerical solution. Diffraction theory is generally applicable to fluid flows around bodies that have characteristic dimensions that are large relative to spatial changes in the fluid field. For example, potential theory generally applies to hydrodynamics of large ships and platforms in moderate-to-low current flow or ocean wave conditions.

On the other hand, viscous theory assumes viscous flow conditions with associated separation of flow (wake formation) at the interface between a boundary layer and the flow. The boundary layer is the thin region of fluid that surrounds the structure. Two hydrodynamic effects result from viscosity, added mass and viscous drag.

Viscous theory is generally applicable to viscid flow around bodies that are slender relative to spatial changes in the fluid field. For example, viscous theory generally applies to hydrodynamics of small, slender bodies such as pile jackets, tubular platforms and cables in moderate-to-high current flow or ocean wave conditions. In particular, viscous theory applies to stationary cylindrical members that meet the following criteria:

$$L/D > 10$$
$$H/D > 1 \qquad (2.23)$$

Where:

D = effective member diameter

H = wave height

L = wave length

Otherwise, diffraction theory applies. For simple bodies and flow, there is a direct relationship between potential and viscous theories (Garrison, 1978).

2.2.2.3 Morison Equation. For viscous unsteady flow resulting from a steady progressive wave, Morison equation (Morison, 1950) is applicable. Originally formulated for computing simple hydrodynamic force on fixed submerged cylinders, engineers now use the Morison equation in extended ways where theoretical adequacy is questionable (Hudspeth and Shields, 1986). These extensions include non-cylindrical cross-sections, frictional fluid forces, and relative motion between the moving fluid and the moving structure (Chakrabarti, 1987). For analytical convenience and tractability, the Morison equation assumes independence between normal and tangential directions and between inertia and drag forces as shown in Figure 2.7.

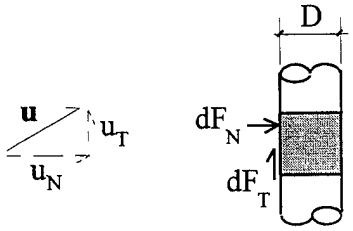


Figure 2.7 Hydrodynamic loads.

Where:

D = effective hydrodynamic diameter of the member

u = velocity of fluid flow (traditional definition)

dF = hydrodynamic force per unit length

N = subscript for component normal to the member

T = subscript for component tangent to the member

Equation (2.24) gives the relative-motion formulation of the Morison equation. This formulation has a vector of force per unit length acting normal and longitudinal to the cylindrical member axis.

$$\mathbf{dF} = \frac{\pi}{4}\rho D^2\left(C_M\dot{\mathbf{u}} - C_A\ddot{\mathbf{x}}\right) + \frac{1}{2}\rho D C_D|\mathbf{u}-\dot{\mathbf{x}}|\,(\mathbf{u}-\dot{\mathbf{x}}) \qquad (2.24)$$

Where:

ρ = mass density of water

C_D = drag coefficient (different in tangential and normal directions)

C_M, C_A = inertia and added mass coefficients

$\mathbf{u}, \dot{\mathbf{u}}$ = fluid velocity and acceleration

$\dot{\mathbf{x}}, \ddot{\mathbf{x}}$ = member velocity and acceleration

Assuming no hydrodynamic interference between members, the total hydrodynamic force is the vector integration of all hydrodynamic and hydrostatic force components on all unit lengths of cylindrical members.

$$\mathbf{R}^M + \mathbf{R}^C = \int_{\Omega} d\mathbf{F} = \text{hydrodynamic force} \tag{2.25}$$

The coefficients in the Morison equation are empirical. For simple cylindrical bodies and ideal flow conditions, fully turbulent drag measurements indicate the following values for the coefficients (Chakrabarti, 1987):

$$\begin{aligned} C_D &= 1.05 \text{ (normal)} \\ C_D &= 0.017\pi \text{ (tangential)} \\ C_A &= 1.0 \\ C_M &= 1 + C_A \end{aligned} \tag{2.26}$$

These coefficients vary dramatically with vorticity, radiation damping, and surface roughness. Experimental measurements are invaluable in this regard. For ideal fluid conditions, hydrodynamic coefficients are generally a function of non-dimensional hydrodynamic parameters (Chakrabarti, 1987).

$$\begin{aligned} \text{Keulegan - Carpenter Parameter} &= u_0 T / D \\ \text{Reynolds Number} &= u_0 D / \nu \\ \text{Diffraction Parameter} &= \pi D / L \\ \text{Surface Roughness} &= e / D \end{aligned} \tag{2.27}$$

Where:

u_0 = maximum magnitude of fluid velocity relative to structure

T = characteristic wave period

ν = kinematic viscosity of fluid

e = characteristic size of roughness particle

The Morison coefficients may also be very dependent on a cyclic phenomenon referred to as vortex shedding In response to high rates of flow, fluid vortices alternately shed off each side of the cylindrical cross-section. The intense structural oscillations that result can cause a profound increase in local drag force.

2.2.2.4 Wind, Current, and Wave Loads. The Morison equation requires that we determine fluid particle velocity and acceleration relative to each structural member. Winds, currents, and waves all contribute fluid flow fields to this definition of fluid particle velocity and acceleration (Zueck, 1986). See Figure 2.8.

Air contributes negligibly to added mass and damping, but at large flow rates may contribute a large static and even dynamic drag force. The majority of the dynamic drag force occurs at very high frequencies, well above structural frequencies. However, wind gusts may vary near structural periods. Very low frequency phenomenon such as tides drive current flow in the ocean. As such, current flow creates a quasi-static drag force.

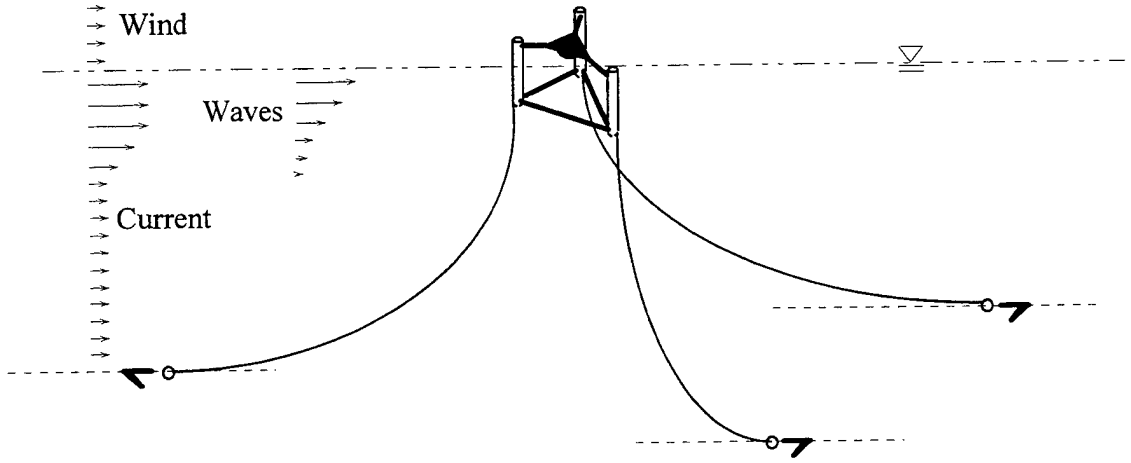


Figure 2.8 Environmental loads on a semisubmersible platform.

Real ocean waves produce flow fields that are quasi-cyclic, multi-directional, and highly nonlinear in nature. We have a myriad of theories that simplify the real wave field (Wiegel, 1964). Most wave theories are unidirectional, periodic and linear. It is extremely hard to compute the resulting fluid kinematics (velocities and acceleration versus depth) for real ocean waves. For lack of a better name, we refer to a field of real ocean waves as a "confused sea."

We can quantify this "confused sea" in a way that is useful for engineering application. Specifically, we can simulate the three-dimensional wave kinematics field for a pre-defined space of fluid around a marine structure (Borgman, et al., 1992). Assuming multi-directional linear wave components, we condition the wave components to specifically adhere to a given set of measured wave spectra and directionality statistics. In addition, we can condition the generated time series to produce a desired time series of kinematics that may be particularly critical to the given structure.

Wind, current, and wave forces may be highly interdependent. For example, current flow affects wave kinematics in a way that is similar to Doppler shift. In addition, wave kinematics depends heavily on local wind influences near the water surface. Hydrodynamic load as defined by the Morison equation is highly dependent on the angle of attack. In addition, wave slamming can generate dynamic force impulses that change rather suddenly in time and space.

2.2.3 Ocean Structural Analysis Codes. Nonlinear ocean structural analysis codes are essentially general purpose, nonlinear finite element codes with all the modeling and solution complications associated with being in the ocean environment.

2.2.3.1 Present Capability. We will presently concentrate on two specific computer codes that are theoretically similar and represent the current state-of-art in ocean structural analysis:

- SEASTAR (PMB, 1989)
- SEADYN (Webster, 1982) and (Webster and Palo, 1989)

SEADYN is specifically for analysis of oceanic cables, whereas SEASTAR is more generally applicable. Written in FORTRAN, both codes have a nonlinear finite element "engine" at their computational core. For SEASTAR, this "engine" is another computer code named ANSR-III (Oughourlian and Powell, 1982). For SEADYN, this "engine" is specific for cable analysis (Webster, 1976).

In our discussions here, we shall concentrate specifically on the cable analysis capability of these programs. The basic finite element used in both codes for modeling cables is the nonlinear bar (or truss) element. The element recognizes large displacement kinematics and equilibrium in the deformed configuration.

SEASTAR and SEADYN use total and updated Lagrangian approaches, respectively, to describe the motion of the system from the current reference state to a new incremental reference state. Both codes offer many options for computing the static and dynamic solution. They also offer numerous strategies for iterative solution including Newton-Raphson iteration, viscous relaxation and line search methods.

One or both codes offer numerous options for controlling the stability of the solution including load or displacement control, variable time step and event-to-event prediction. The reason for the many options is that no single option offers a robust and stable way of computing the solution for the entire intended class of structural problems.

Specific numerical representations provide for ocean-related loads including gravity, hydrostatic pressure, buoyancy, added-mass, fluid-induced damping and wave- or current-induced hydrodynamic drag and inertia. Many of these load representations are highly nonlinear and orientation specific.

Loads fall into three separate types:

- Dead - Loads that have no temporal variation, such as weight, hydrostatic pressure, and buoyancy.
- Live - Temporal loads that produce insignificant acceleration, such as water currents, wind, and various operational loads.
- Dynamic - Temporal loads that produce significant acceleration, such as waves, vessel motion, and earthquake motions.

Given these three types of loads, SEASTAR or SEADYN assume a three-phase approach to dynamic analysis as shown in Figure 2.9. The static phase for the dead loads must begin from

an initial state that is close to force equilibrium. The static phase for the live load uses the final configuration from the dead load as its "initial state."

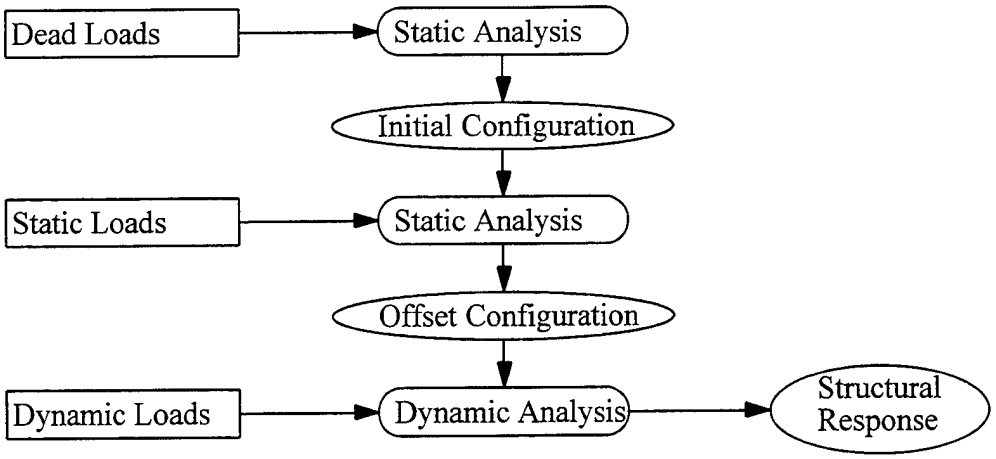


Figure 2.9 Three-phase analysis procedure.

We input static loads as patterns and dynamic loads as time series of records. Each analysis segment is a combination of load patterns and records. If desired, we can add new patterns and records in any analysis segment. Numerical solution strategies can differ for each of the three phases.

2.2.3.2 Deficiencies in Present Capability. The physical modeling capabilities of SEADYN and SEASTAR are adequate for analyzing the majority of oceanic cable problems. However, there is always the need for further modeling enhancements, such as better hydrodynamic interaction, better material properties for synthetic lines and better representation of geotechnical interaction. Major difficulties in the computational framework of both codes hamper and even preclude better physical modeling theories.

Although SEADYN and SEASTAR possess numerous strategies for nonlinear solution, both codes exhibit slow or non-convergent nonlinear solutions. This is particularly true when the analysis includes many nonlinear modeling complexities.

In a slack state, the traditional truss element often has inappropriate resistance. As such, traditional truss elements easily respond with wholly unrestrained motion, resulting in solution instability. In contrast, real cables always respond within an ultimate stable bound. This bound is the straight, stretched, unbroken length of the cable.

Dynamic analysis is less prone to solution inefficiencies and instability than static analysis because damping and inertia terms help condition the numerical solution process. Therefore, the most pressing computational problem is with static analysis.

One particular source of frustration is determining the initial static shape of a cable structure (Webster, 1980). One cannot prescribe the initial shape and the prestress force independently of force equilibrium as in most other structural analysis. Since cable systems derive their stiffness and thus their ability to support loads from proper preloading, it is necessary to provide accurate estimates of initial shape and tensions. If the initial tension estimate is not

consistent with the initial shape estimate, the structure stiffness matrix will be ill-conditioned and the solution will often not converge.

Required methods for getting a good initial starting state include building the cable structure in some highly distorted way using imaginary constraints or loads that assure well-conditioned initial stiffness. The distorted shape then evolves into the desired shape by adding real loads or constraints and removing the imaginary ones. For a complex three-dimensional cable structure, this procedure is unworkable.

It is desirable to analyze marine cable systems with extremely low tensions. As such, the cable system has very little if any stiffness, and the numerical model representing the cable system can easily develop a system singularity with no possible solution. In addition, the many deficiencies in the modeling of fluid flow interaction become apparent when a cable approaches zero tension.

2.3 Substructuring Concepts

A local/global computational approach is an extension of traditional concepts for substructuring. We choose to classify traditional substructuring concepts as either matrix condensation, matrix decomposition, or multi-domain methods.

2.3.1 Matrix Condensation. Static matrix condensation reduces the size of large structural systems (Prezemieniecki, 1963). In matrix notation, static matrix condensation is a formal procedure for removing matrix terms representing unwanted degrees of freedom, leaving a smaller linear system for final solution. Rather than simply dropping the unwanted degrees of freedom, this procedure factors the structural influence of the unwanted into the desired degrees of freedom.

Mathematicians refer to static matrix condensation as formulating a Schur complement (Duff, 1989). Numerically, matrix condensation is simply a reordering of the computations necessary for normal Gaussian elimination. Solving for all unknowns using either normal or partitioned Gaussian elimination requires the same total number of non-trivial mathematical operations. In other words, for a linear system, there is no inherent computational savings in using static matrix condensation (Williams, 1973), unless there are several exactly repeatable substructures.

Solving for eigen-functions can be a very computationally intensive exercise (Parlett, 1980). Therefore, dynamic matrix condensation methods exist for reducing the required set of eigen-functions and thus reducing the size of the computational task. Most of these matrix condensation methods are variations of the classical Guyan condensation technique (Guyan, 1965). Component mode synthesis is a class of Guyan condensation techniques that specifically deal with large, sparse dynamic structural systems (Dickens, 1980). Component mode synthesis recognizes and exploits the obvious relationship between significant eigen-functions (principal modes of motion) and physically identifiable components of the structure.

An error exists between the reduced and true modes of vibration. There are many variations of component mode synthesis that attempt to minimize this error (Craig, 1995). In the early years of computer-based structural analysis, computers had limited storage, so engineers made great use of matrix condensation simply to reduce the in-core storage requirements for the computational problem. This is no longer necessary.

2.3.2 Matrix Decomposition. The technical literature is full of numerical methods for decomposing a single large sparse matrix representing a huge system into smaller, denser submatrices. These matrix decomposition methods differ in how they identify highly coupled dense clusters of the single large sparse matrix. These purely mathematical methods also differ in how they partition or modify these clusters into smaller, denser submatrices. The degree of coupling between submatrices differs between methods.

Matrix partitioning is a method that simply divides the large matrix into smaller submatrices with rearrangement of rows and columns as necessary. Matrix partitioning maintains full system representation and solution accuracy.

Branch tearing is a method that separates the large sparse system into dense submatrices by eliminating sparse matrix terms. These sparse matrix terms represent the branches that loosely connect the various submatrices. This method recognizes the low-rank connection between submatrices and modifies the large sparse system matrix into several independent or uncoupled submatrices.

Nodal tearing is a method that separates the large sparse system into dense submatrices by creating separate independent submatrices representing the branches and their nodes (Farhat and Roux, 1991). This method preserves full system representation.

2.3.3 Multi-Domain Methods. Multi-domain methods (or domain decomposition methods) exist for solving general partial differential equations (Keyes, 1991). The optimal choice of a specific model representation for any given sub-domain of the problem can, in principle, include any of the following diverse choices:

- Partial differential equation, e.g., potential, thermodynamic or structural
- Spatial discretization, e.g., structured or unstructured grids
- Spatial approximation, e.g., finite difference, finite element or spectral
- Order of spatial approximation, e.g., 2nd order with p- or h-refinement
- Temporal approximation, e.g., Runge-Kutta, Wilson-Theta, or trapezoidal rule
- Time stepping scheme, e.g., explicit, implicit, or semi-implicit
- Order of time stepping, e.g., 1st, 2nd, or 3rd, or even variable

A general review of time integration schemes with an emphasis on those pertaining to semi-discretization of multi-physics systems is available (Belytschko, 1989). Topics include linear multi-step methods, partitioning, and sub-cycling. These schemes work well for linear fluid/structure interaction that couple the equations of fluid acoustics and structural dynamics.

A partitioned integrator uses different time integration methods for solving different parts of the finite element mesh. Schemes for implicit/explicit node partitions, implicit/explicit element partitions, and implicit/implicit staggered partitions are available (Park and Felippa, 1983). The motivation here is to use the stability of an implicit integrator on partitions sensitive to stability and the computational efficiency of an explicit integrator on partitions insensitive to stability. Staggered methods allow local solutions to lag behind the global solution.

Adaptive techniques for refinement of finite mesh size and element order in various sub-domains help improve both solution speed and accuracy (Oden and Demkowicz, 1989).

Algorithms based on pre-conditioned conjugate gradient offer a convenient solution for full nonlinear dynamic simulation of substructured continuum problems (De Roeck, 1992).

There are many variations of domain-partitioned time integration schemes for solving the linear dynamic structural problems (Wood, 1987). There are no such methods applicable to the nonlinear case. In their concluding remarks, Hajjar and Abel suggest the following alternative method to their own: "... full substructuring with condensation of internal degrees of freedom followed by an interface solution" (Hajjar and Abel, 1989). This is precisely the basis for the local/global computational framework that we formulate in Chapter 3.

3. IMPROVED MODELING THEORIES

A local/global approach to modeling large structural systems requires a new two-level computational framework. Not only must the framework allow for different local submodels, but it must also provide for effective global integration of all submodels. This requires careful choice of modeling theories both at the local and global levels.

Section 3.1 introduces the preferred local/global modeling choices and describes the relationship between submodels and super-elements. Section 3.2 develops a semi-analytical cable super-element. Section 3.3 describes an evolutionary new finite cable element and Section 3.4 uses this element to create a revolutionary new numerical cable super-element. Sections 3.5 and 3.6 describe other unique elements for use in local/global simulation of some marine compliant structures.

3.1 Modeling Choices

There are many choices for modeling a large and complex structural system. We concentrate on modeling choices that allow for a local/global approach. Good modeling practice requires careful attention to proper time and space scales.

3.1.1 Scales of Time and Space. For accurate physical representation of a given problem in structural dynamics, the structural analyst must focus on models with the proper scale of time and space. For compliant marine structures, there are often three classically recognized scales of time and space (each dominated by a particular type of load):

1) Time period greater than 30 seconds and length scale on the order of water depth (tidal, current, wind and second-order wave load)
2) Time period of 30 to 2 seconds and length scale on the order of element length (first-order wave load)
3) Time period less than 2 seconds and length scale on the order of element thickness (fluid vortex load)

In recognition of these three domains, structural analysts often create three different models to compute the structural response in each scale of time and space:

1) A quasi-static model for tidal and current force acting on the overall structure

2) A linear dynamic model for wave forces acting on individual members
3) A fluid-structure interaction model for fluid forces acting on finite surfaces

In reality, we cannot so easily divide the domains. For example, low-frequency tidal velocities may suddenly cause high-frequency vortex-induced vibrations. It is not hard to pose other cases where the nonlinear structural models must span numerous scales of time and space in order to give an adequate picture of the structural response. Theoretically, the entire spectrum of scales in time and space may be important. For example, wave slam is an impulsive load with a very small scale of time.

Structural analysts often limit their analysis to one discrete scale of time and space. With this single domain approach, the structural analyst uses one large finite element model for the entire structure. For a linear structural system, this single domain approach works well. For a nonlinear structural system with changing nonlinear behavior, we often must re-linearize and re-solve the complete structural system at each step or substep of the nonlinear simulation. Local inefficiencies, numerical instabilities and modeling errors may develop in a particular substructure and cause general nonlinear simulation inaccuracy or instability. As such, this single domain approach is often unworkable for nonlinear structural systems.

3.1.2 A Nonlinear Local/Global Approach. To improve nonlinear simulation, we choose two scales of time and space, namely a local and a global domain. A local model represents the local physics of each local substructure. Likewise, the global model represents the integrated physics of the overall structure. In principle, each submodel can have its own local scale of time and space. For simplicity in the current research, we choose to keep the same scale of time for all submodels.

To simulate the global response of a structure, we compute the local responses of each submodel. We then condense each local submodel to a global super-element. Made up of the assembly of all super-elements, the global model thus represents the integrated physics of all submodels.

We can develop submodels specialized for the local nonlinear physics of each substructure. We can also develop nonlinear solution strategies specialized for each submodel. As such, we can balance the local complexity of each submodel with the overall nonlinear solution efficiency.

A local/global modeling approach can concentrate computational power "where" and "when" needed. The "where" means that we choose submodels based on a spatial requirement. If we expect a particular part of the structure to cause highly nonlinear behavior, then we direct appropriate submodels at stabilizing this spatial nonlinearity. The "when" means we choose submodels based on a temporal requirement. If we expect highly nonlinear behavior at a particular time step, then we activate or de-activate different submodels for each substructure as needed to meet this temporal nonlinearity.

As an example, compliant marine structures often consist of many interconnected cables, each a potential substructure. If a particular cable substructure has an in-line buoy or an in-line weight that causes the cable to take a very different spatial shape, then we may choose to subdivide further. If a particular cable substructure touches the seafloor at a particular time step, we may choose to switch to a higher level cable/seafloor submodel for this particular interaction.

Figure 3.1 shows the interaction between the submodels and the global model. For each step of the overall simulation, the figure depicts a global substep loop for solution of the global model. For each global substep, the figure depicts a local substep loop for solution of each submodel.

First we establish a starting configuration for the global model. Then, for each step of the simulation, we distribute global configuration information to each submodel. Given local configuration information, we check local force equilibrium in each submodel relative to the desired force equilibrium tolerance. We solve only those submodels with significant local force imbalance using as many local substeps as required.

As each local solution converges to local force equilibrium, we condense each submodel to a corresponding super-element. The super-element results from condensing out the local nodes of each submodel in favor of the global nodes.

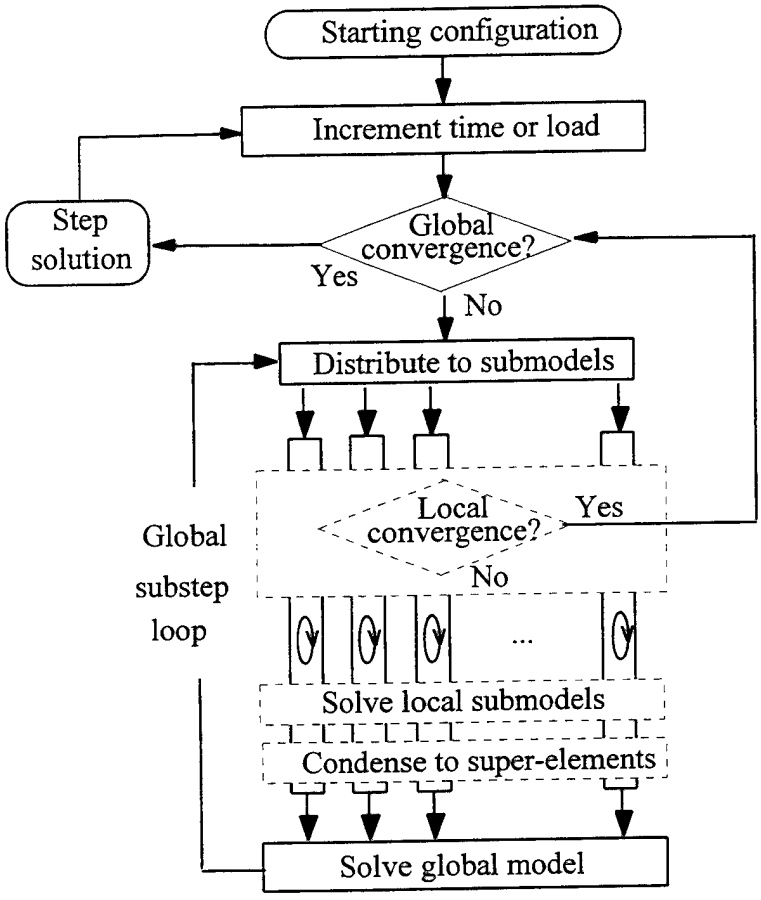


Figure 3.1 Local/global solution stepping.

We choose a local/global designation of nodes over the traditional internal/external designation to avoid a probable confusion. Namely, a node internal to one submodel is external to another submodel. We prefer to have a designation that applies universally across the entire

model. Therefore, we designate a node internal to any submodel as a local node and a node external to all submodels as a global node. Most analysts (including the dissertation chair) prefer the traditional internal/external designation of nodes as this allows for multiple levels of substructuring. In this dissertation we deal with only two levels--a local and a global domain.

Using the traditional direct stiffness method of structural analysis, we assemble all super-elements (and any regular elements at the global level) into a global model. If the local and global coordinate systems are not equivalent, we must transform from local to global coordinates. Given a complete set of boundary conditions, we solve the assembled linear algebraic equations for all global displacements in the global substep. We then go to the next global substep and again redistribute global data to the submodels.

Whereas the local nodes represent the complete behavior of the submodel at the local level, the global nodes represent the complete behavior of its super-element at the global level. Reduced to the global nodes, the super-element is effectively a linearization of the submodel for the global substep. The global model is thus piece-wise linear between global substeps.

The strongest nonlinear behavior in most structures often occurs only in a specific substructure. To maintain nonlinear solution stability, this local nonlinear behavior may require a special nonlinear solution strategy. For example, surface contact may require very small sub-increments of contact load for stable simulation of local contact deformation. In a traditional one-domain model, we must then apply this special nonlinear solution strategy across the entire structural system, resulting in numerous trivial computations in parts of the structural system far from the contact surface.

The beauty of a local/global computational framework is that it concentrates local nonlinear behavior in submodels. How each submodel solves the specific peculiarities of its nonlinear behavior can remain locally private to the submodel. The submodel can subdivide the global substep into as many local substeps as required to affect a final solution for the global substep. The global nonlinear solution strategy need not concern itself directly with highly nonlinear behavior in the submodels. This relieves the global model of these highly nonlinear solution requirements.

A major computational advantage of a local/global approach is that we only need to linearize and solve those submodels that change substantially in behavior. If a particular submodel behaves quite linearly, then a constant super-element stiffness is sufficient for the next substep of the global solution. Consequently, we need not necessarily solve nor condense every submodel at every global substep.

Super-element events, based on motions of only super-element nodes, are useful for deciding whether to solve a submodel. If a global solution increment triggers a super-element event (or even just a force imbalance threshold), we solve the submodel. If not, we use the old super-element properties.

With a local/global computational framework, we can allow for a very large breadth of local physical behavior without unduly burdening the robustness of the overall simulation task. We generalize the global nonlinear solution strategy so that it can integrate a large breadth of potential super-elements.

3.1.3 Cable Submodels and Super-Elements. The first step for creating a viable local/global computational framework involves developing robust submodels. Since cables are the substructures that connect most other marine substructures, we concentrate on developing

specialized cable sub-models. In these sub-models, we focus on a particularly important behavior of real cables, namely the ability to snap from slack-to-taut and vice versa.

For very flexible cable materials (e.g., rubber bungy cord), the difference between slack and taut behavior is very indistinct. For very stiff cable materials (e.g., wire rope), the difference between slack and taut behavior is very distinct. We demonstrate the importance of slack/taut behavior with typical cable seen in Figure 3.2.

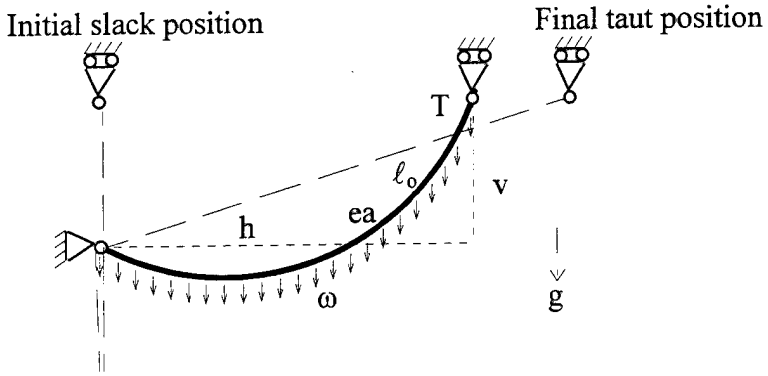


Figure 3.2 Typical cable for showing slack/taut behavior.

Where:

ea = $3x10^4$ lb = stiffness factor

ω = 1 lb / ft = weight per unit length

ℓ_0 = 100 ft = unstretched length of entire cable

v = 60.0 ft = vertical dimension

g = 32.2 ft / s^2 = acceleration due to gravity

T = tension at top of cable

h = horizontal displacement

Figure 3.3 shows the distinction between slack/taut behavior for the typical cable given in Figure 3.2.

In Figure 3.3, the cable is slack or taut for a horizontal displacement less than or greater than 80 feet, respectively. Catenary behavior is important when the cable is slack, and elastic stretching behavior is important when the cable is taut. The change from slack-to-taut can cause large dynamic impulsive forces. For realistic representation, cable submodels must also possess this distinct slack/taut behavior.

Concentrating on correct catenary behavior, we develop a simple semi-analytical cable super-element based on elastic catenary expressions. Concentrating on correct elastic stretching behavior, we develop a more capable numerical cable super-element based on a new finite cable element.

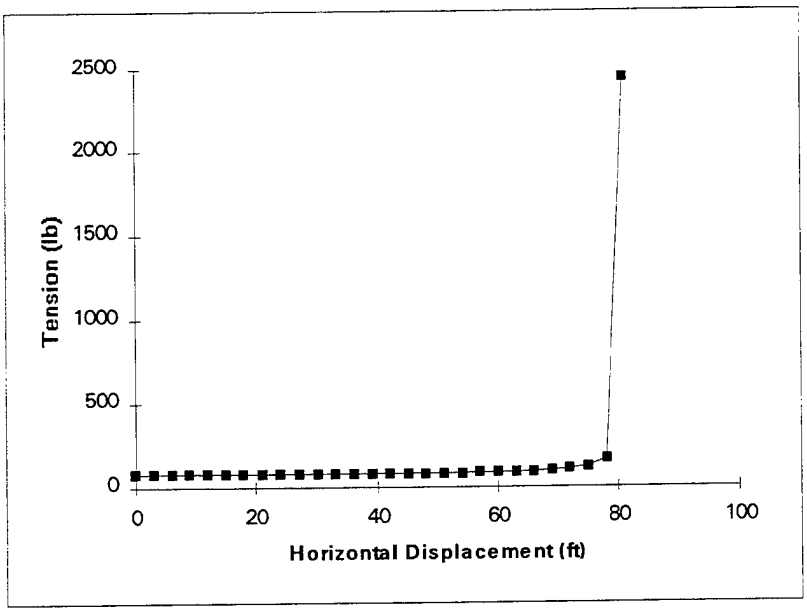


Figure 3.3 Slack/taut behavior of a typical cable.

3.2 Semi-Analytical Cable Super-Element

We derive a fairly simple static cable super-element from basic hyperbolic expressions representing the catenary shape of a cable under the influence of a uniform field load. Under a uniform field load such as gravity, cable catenaries are essentially planar in shape. The two ends of the cable plus the field load vector define this plane. Figure 3.4 depicts the desired cable super-element.

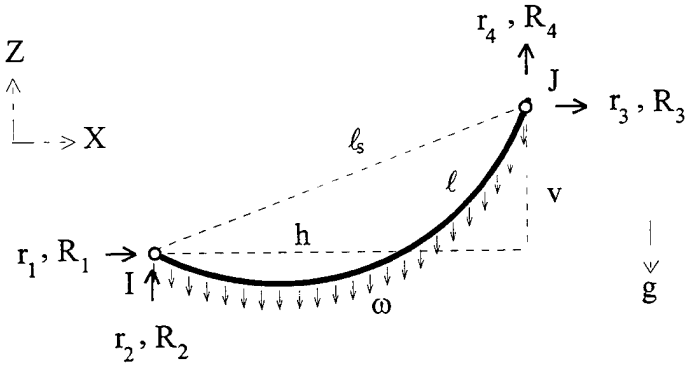


Figure 3.4. Semi-analytical cable super-element.

Where:

h, v = catenary dimensions

I, J = super - element nodes

ℓ = cable length

ℓ_s = super - element (chord) length

$\mathbf{r}, \mathbf{R}$ = displacement, force for super - element nodes

ω = load per unit length

X,Z = global coordinates

g = direction of field load

The super-element axis is a line drawn from node "I" to node "J." The cable supports only tensile forces.

3.2.1 Simple Elastic Catenary Theory. Development of the cable super-element begins with the basic hyperbolic catenary expressions for the inelastic catenary presented in section 2.2.1.1. To get a cable super-element that takes a catenary shape when slack and stretches straight when taut, we add linear rope elasticity to the basic catenary expressions. Integrating along the elastic cable, we obtain the following alternative geometric relationships (Peyrot and Goulois, 1979).

$$h = -R_1\left(\frac{\ell_u}{E_0 A_0} + \frac{1}{\omega} log \frac{R_4 + T_J}{T_I - R_2}\right) \tag{3.1}$$

$$v = \frac{1}{2E_0 A_0 \omega}\left(T_J^2 - T_I^2\right) + \frac{T_J - T_I}{\omega} \tag{3.2}$$

$$\ell = \ell_u + \frac{1}{2E_0 A_0 \omega}\left(R_4 T_J + R_2 T_I + R_1^2 \, log \frac{R_4 + T_J}{T_I - R_2}\right) \tag{3.3}$$

Where:

A_0 = initial cross - sectional area of cable

E_0 = initial modulus of elasticity in line with cable

ℓ_u = unstressed cable length

T = tension in cable

Using Figure 3.4, we write additional equations of statics to reduce the number of unknowns.

$$\begin{aligned} R_4 &= -R_2 + \omega \ell_u \\ R_3 &= -R_1 \\ T_I &= \sqrt{R_1^2 + R_2^2} \\ T_J &= \sqrt{R_3^2 + R_4^2} \end{aligned} \tag{3.4}$$

Combining equations (3.1) through (3.4), we get complex nonlinear expressions for the catenary dimensions that are implicit functions of the end forces at node "I."

$$\begin{aligned} h &= \Im_h\left(R_1, R_2\right) \\ v &= \Im_v\left(R_1, R_2\right) \end{aligned} \tag{3.5}$$

3.2.2 Local Submodel Solution. We choose an iterative procedure to solve equation (3.5). Equation (3.6) gives an appropriate set of starting values for this iterative procedure. The sign of these starting values assumes that node "I" is lower than point "J." If not, switch the two nodes.

$$\begin{aligned} \left(R_1\right)_0 &= -\frac{\omega h}{2\lambda_0} \\ \left(R_2\right)_0 &= \frac{\omega}{2}\left(-v\frac{cosh(\lambda_0)}{sinh(\lambda_0)} + \ell_u\right) \end{aligned} \tag{3.6}$$

Where:

$$\begin{aligned} \lambda^0 &= \sqrt{3\left(\frac{\ell_u^2 - v^2}{h^2} - 1\right)} && \text{for } \ell_u \geq \ell_s \text{ and } h \neq 0 \\ \lambda^0 &= 0.2 && \text{for } \ell_u < \ell_s \text{ and } h \neq 0 \\ \lambda^0 &= 10^6 && \text{for } h = 0 \end{aligned}$$

Often the bottom portion of a cable catenary must lie along and conform to the seafloor. We can use another iterative loop within the normal iterative loop to enforce geometric compatibility between the cable and the seafloor (Webster, 1982).

3.2.3 Conversion to Super-Element. To develop an incremental super-element representation of the cable submodel, we next derive a flexibility relationship for the super-element forces and displacements at node "I." We do this by fixing node "J" and taking the partial derivatives of equation (3.5) relative to the displacements at node "I."

$$\begin{bmatrix} dh \\ dv \end{bmatrix} = \mathbf{f}\begin{bmatrix} dR_1 \\ dR_2 \end{bmatrix} \equiv \begin{bmatrix} f_{h1} & f_{h2} \\ f_{v1} & f_{v2} \end{bmatrix}\begin{bmatrix} dR_1 \\ dR_2 \end{bmatrix} \tag{3.7}$$

Where:

$$f_{h1} = \frac{\partial \Im_h}{\partial R_1} = \frac{h}{R_1} + \frac{1}{\omega}\left(\frac{R_4}{T_J} + \frac{R_2}{T_I}\right)$$

$$f_{h2} = \frac{\partial \mathfrak{I}_h}{\partial R_2} = \frac{R_1}{\omega}\left(\frac{1}{T_J} - \frac{1}{T_I}\right)$$

$$f_{v1} = \frac{\partial \mathfrak{I}_v}{\partial R_1} = f_{h2}$$

$$f_{v2} = \frac{\partial \mathfrak{I}_v}{\partial R_2} = \frac{-\ell}{EA} - \frac{1}{\omega}\left(\frac{R_4}{T_J} + \frac{R_2}{T_I}\right)$$

Inverting equation (3.7) and expanding to include both super-element nodes, we get an incremental stiffness representation for a two-dimensional, elastic cable super-element.

$$\mathbf{K}_n = \begin{bmatrix} \mathbf{f}^{-1} & -\mathbf{f}^{-1} \\ -\mathbf{f}^{-1} & \mathbf{f}^{-1} \end{bmatrix} \tag{3.8}$$

To extend the stiffness matrix to three dimensions requires consideration for motion out of the current catenary plane.

3.3 Finite Cable Element

For more complex behavior, we prefer to discretize the cable substructure into a submodel of several serially connected finite elements. A condensation procedure converts this submodel into a cable super-element. However, we must first develop a finite cable element that promotes stable solution of the cable submodel.

We choose a finite cable element that has two nodes with three translational degrees of freedom in each coordinate direction, as shown in Figure 3.5. The element can take any orientation in the X-Y (2-D) or X-Y-Z (3-D) plane.

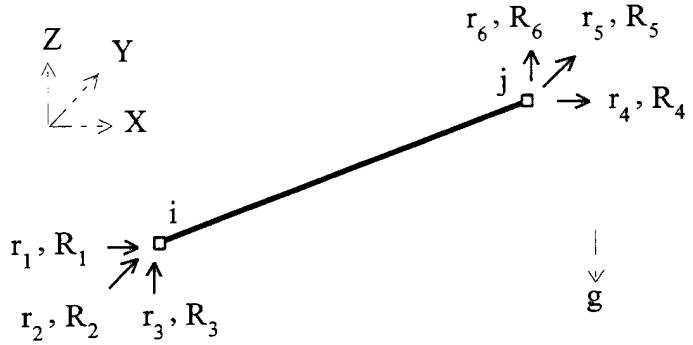


Figure 3.5 Finite cable element.

Where:

$\mathbf{r}, \mathbf{R}$ = displacement, force for nodes of element

i, j = element nodes

X, Y, Z = global coordinates

g = direction of gravity acceleration field

3.3.1 Element Data and State. For development of the new finite cable element, we use the following unique set of cable modeling principles:

- Allow initial nodal coordinates to be just a convenient place to start.
- Choose element length to be the distance between element nodes.
- Select element deformation to be simply element length.
- Define undeformed element length to be zero (i.e., coincident nodes).
- Base degrees of freedom on global coordinates.
- Store global coordinates instead of displacements.

To track response, we need three distinct types of data for the new finite cable element:

- Fixed properties (e.g., unit weight): These data define the basis of the element and remain constant throughout the simulation.
- Variable properties (e.g., slack/taut element length): These data redefine the element and may change at the beginning of each step of the simulation.
- State variables (e.g., element deformation): These data define the element state and can change at each substep of each step of the simulation.

For the finite cable element, we define only three general state variables, namely element deformation, action, and directional cosines. We define element deformation and action to be element length and tension, respectively. Element length (and thus element deformation) is the distance between the two nodes of the element along the element axis. Given any nodal positions, element deformation and action are always uniquely positive, non-zero values. Consequently, this element supports only tension.

Furthermore, we define only one variable property for the finite cable element, namely its slack/taut length. We define slack/taut length to be the natural elemental length of the cable when it hangs vertically under the influence of gravity. Since the bottom end of a hanging cable has zero tension, this length is also the natural definition for when a cable is either slack or taut, hence the name, "slack/taut" length.

The difference between current element length and the specified "slack/taut" length determines whether the cable element is taut or slack. Because of this absolute choice, initial element position has no special significance to the final simulation answers. We can choose to make any element either initially taut or initially slack simply by assigning a slack/taut length that is smaller or larger than the initial distance between nodes. Preferably, this slack/taut length represents the true length of the cable. Length only has a unique meaning after applying some

level of tension to straighten out the cable. Consequently, the slack/taut length has a small tension associated with it.

How does this new definition of state differ from the traditional cable model that uses "truss" elements to model cable behavior? For a "truss" element, deformation is the change in length from a specified "zero-tension" length. Traditional "truss" elements differ in what to do when element length becomes less than this assumed "zero-tension" length. Some models remove the element, while others provide compressive resistance. Both assumptions are physically incorrect for any cable of appreciable length. Traditional "truss" elements are thus ill-advised for representing real cables.

We require the following key ingredients for full nonlinear formulation of our finite cable element:

- Small strains that are consistent with rotated engineering strain kinematics
- Elastic rope as the basic material constitutive relationship
- Force equilibrium at the deformed configuration
- Geometric nonlinearity for kinematics and force equilibrium

We develop the governing equations describing structural behavior of the finite cable element from element-based relationships for kinematic continuity, action-deformation, and force equilibrium.

3.3.2 Kinematic Continuity Relationship. A kinematic continuity relationship ensures compatibility between element deformation, element rotation and nodal displacement. Given previous nodal coordinates and an increment of displacement, we desire to compute element deformation and rotation. Since the undeformed length of the finite cable element is zero, all classical strain measures are undefined for the finite cable element. As a consequence, we choose to use simple element length as the measure of element deformation. We also use a rotated engineering reference frame for defining element deformation of the cable (Argyris and Scharf, 1972). This reference frame is similar to a co-rotational reference frame (Crisfield, 1991). In contrast, the popular Lagrangian reference frame would unnecessarily complicate the simple nature of the cable finite element and cause serious numerical problems as element length (element deformation) approaches zero.

We express all physical variables for the current incremental state of the element in terms of the previous state and an incremental displacement of the two nodes. In moving incrementally from the previous to the current state, the rotating reference frame separates total element deformation from incremental element rotation as shown in Figure 3.6. Since lines through the current and previous states do not necessarily intersect, the vector of incremental element rotation consists of the angles between the lines projected into the three component planes.

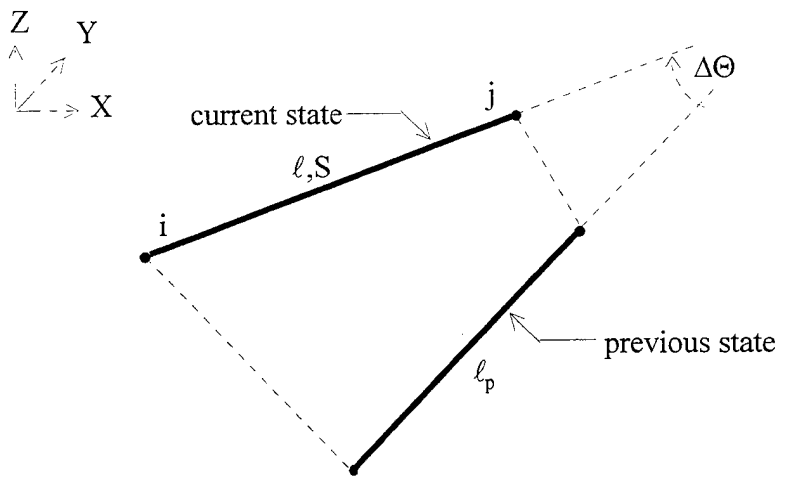


Figure 3.6 Rotated engineering formulation.

With a rotating reference frame, it is convenient to represent element kinematics using updated global coordinates, directional cosines and incremental element rotation. We base all kinematic computations on the global coordinates of the end nodes.

$$
\begin{aligned}
\ell &= |\mathbf{c}'| = \text{current element length} \equiv \text{current axial deformation} \\
\ell_p &= \text{element length at previous state} \\
\Delta\Theta &= \begin{bmatrix} -\bar{\mathbf{u}} \\ \bar{\mathbf{u}} \end{bmatrix} = \text{incremental element rotation} \\
S &= \text{element action (tension)}
\end{aligned}
\tag{3.9}
$$

Where:

$$
\mathbf{c}' = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}
$$

$\mathbf{x} = \mathbf{x}_p + \Delta\mathbf{r}$ = coordinates of element ends

$\mathbf{x}_p = \begin{bmatrix} x_i & y_i & z_i & x_j & y_j & z_j \end{bmatrix}_p^T$ = previous coordinates of element ends

$\Delta\mathbf{r}$ = incremental displacement of element ends

$\bar{\mathbf{u}} = \hat{\mathbf{c}}\left(\mathbf{u}_j - \mathbf{u}_i\right)$ = incremental transverse displacement (normalized)

$\hat{\mathbf{c}} = \mathbf{I} - \mathbf{c}\,\mathbf{c}^T$

$$
\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$\mathbf{c} = \frac{1}{\ell}\mathbf{c}' = \text{directional cosines of element axis}$$

$$\mathbf{u}_i = \begin{bmatrix} \Delta r_1 & \Delta r_2 & \Delta r_3 \end{bmatrix}^T$$

$$\mathbf{u}_j = \begin{bmatrix} \Delta r_4 & \Delta r_5 & \Delta r_6 \end{bmatrix}^T$$

Matrices for transformations from nodal displacement to element deformation and rotation depend upon updated coordinates. Therefore, measures of element deformation and rotation vary as full nonlinear functions of incremental or infinitesimal displacement.

$$\begin{aligned} \Delta\ell &= \text{finite real axial deformation} \approx \mathbf{a}\ \Delta\mathbf{r} \\ d\ell &= \text{infinitesimal real axial deformation} = \mathbf{a}\ d\mathbf{r} \\ d\Theta &= \text{infinitesimal real geometric rotation} = \mathbf{b}\ d\mathbf{r} \end{aligned} \tag{3.10}$$

Where:

$$d\mathbf{r} = \text{infinitesimal real displacement for nodes of element}$$

$$\mathbf{a} = \begin{bmatrix} -\mathbf{c}^T & \mathbf{c}^T \end{bmatrix} = \text{axial transformation}$$

$$\mathbf{b} = \begin{bmatrix} \hat{\mathbf{c}} & -\hat{\mathbf{c}} \\ -\hat{\mathbf{c}} & \hat{\mathbf{c}} \end{bmatrix} = \text{geometric transformation}$$

3.3.3 Action-Deformation Relationship. An action-deformation relationship defines the constitutive relationship between element action and element deformation. A cable element should behave like the cable catenary that it models. For a typical cable, axial and rotational stiffness predominates. Stiffness in shear and bending is generally insignificant.

Torsional stiffness may be significant. For example, a cable under low tension may exhibit a torsional instability whereby torsional stress causes the cable to twist out of plane locally. Known as hockling, this condition develops from coupled axial and torsional stiffness (Rosenthal, 1976). However, if properly designed with torque-balanced cross-sections, cables of any appreciable length exhibit little torsional response.

Consequently for development of the finite cable element, we choose to ignore all bending, shear, and torsional effects. If warranted, we can include these other effects in a more complex version of the finite cable element.

We assume a bi-linear elastic relationship between element action and axial deformation as shown in Figure 3.7. When slack, the element has a very small axial stiffness. When taut, the element has a very large axial stiffness. For typical wire rope type cables of appreciable length, the ratio of taut to slack stiffness is often 10,000. The chosen action-deformation relationship (shown in Figure 3.7) closely resembles the relationship between tension and extension for a typical cable (shown in Figure 3.3).

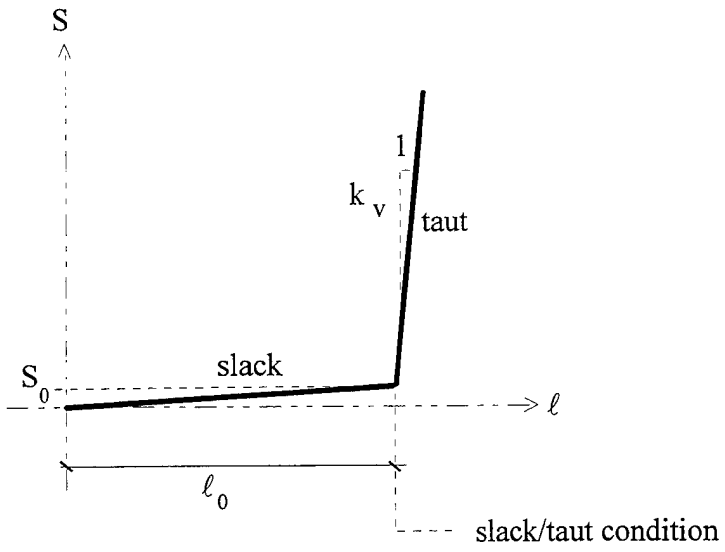


Figure 3.7 Action-deformation relationship.

The infinitesimal relationship between axial action and deformation is bi-linear. The relationship could just as easily be multi-linear.

$$dS = k\, d\ell \tag{3.11}$$

Where:

$$k = \left\{ \begin{array}{l} k_s = {S_0}/{\ell_0} = \text{minimal axial stiffness} \\ k_v = {A_0 E_0}/{\ell_0} = \text{axial material stiffness} \end{array} \right\} \text{ for } \left\{ \begin{array}{ll} \ell < \ell_0 & \text{(slack cable)} \\ \ell \geq \ell_0 & \text{(taut cable)} \end{array} \right\}$$

dS = infinitesimal real element action (tension)

S_0 = assigned slack / taut tension (e.g., weight of element)

A_0 = element cross – sectional area in slack / taut condition

E_0 = Young's elastic modulus in slack / taut condition

ℓ_0 = element slack / taut length

Cable action is always positive (tension). Hence, stiffness is always positive, whether the cable is slack or taut.

3.3.4 Force Equilibrium Relationship. A force equilibrium relationship relates nodal force to element action. We impose virtual infinitesimal displacements in the rotated and deformed configuration to establish full nonlinear force equilibrium and get internal forces in the current state.

$$d\hat{\mathbf{r}}^T\mathbf{R} = d\hat{\ell}^T S = d\hat{\mathbf{r}}^T\mathbf{a}^T S \tag{3.12a}$$

$$\mathbf{R} = \mathbf{a}^T S \tag{3.12b}$$

Where:

$d\hat{\mathbf{r}}$ = infinitesimal virtual displacement for nodes of element

$d\hat{\ell}$ = infinitesimal virtual element deformation

3.3.5 Linearization. For incremental force equilibrium, we differentiate equation (3.12b).

$$d\mathbf{R} = d(\mathbf{a}^T S) = \mathbf{a}^T dS + d\mathbf{a}^T S = \text{infinitesimal nodal force} \tag{3.13}$$

Where (Argyris & Scharf, 1972):

$$d\mathbf{a}^T = \frac{1}{\ell} d\Theta$$

We linearize the stiffness for each element by calculating a tangent stiffness for the current state. Making proper substitutions, we get the following relationships for axial (elastic) and rotational (geometric) stiffness.

$$d\mathbf{R}_E \equiv \mathbf{a}^T dS = \mathbf{a}^T k\, \mathbf{a}\, d\mathbf{r} \equiv K_E\, d\mathbf{r} \equiv \begin{bmatrix} K_E' & -K_E' \\ -K_E' & K_E' \end{bmatrix} d\mathbf{r} \tag{3.14}$$

$$d\mathbf{R}_G \equiv d\mathbf{a}^T S = \frac{S}{\ell} d\Theta = \frac{S}{\ell}\, \mathbf{b}\, d\mathbf{r} \equiv K_G\, d\mathbf{r} \equiv \begin{bmatrix} K_G' & -K_G' \\ -K_G' & K_G' \end{bmatrix} d\mathbf{r} \tag{3.15}$$

Where:

$K_E' = k\, \mathbf{c}\, \mathbf{c}^T$ = tangent axial (elastic) stiffness

$K_G' = -\frac{S}{\ell}(\hat{\mathbf{c}})$ = tangent rotational (geometric) stiffness

In the slack condition, rotational stiffness remains constant even as the element length approaches zero. As such, element stiffness is always well conditioned for any displacement of the element nodes. In other words, we need no artificial stiffness terms to pre-condition the finite cable element.

3.3.6 Element Events. An element event is anything occurring within the element, such as yielding, nonlinear unloading, gap closure, etc., that may significantly change the tangent stiffness or affect the force imbalance. For the finite cable element, we choose an axial and a

rotational event. With the necessary independence between element axial deformation and geometric rotation, a rotated engineering formulation allows for independent definition of these two types of element events.

We choose two possible axial events, namely the finite cable element going from taut to slack and vice versa. These two events represent the discrete change in axial stiffness as shown in Figure 3.7. To avoid a large force imbalance, we must precisely compute the axial events, as described in Chapter 4.

Since element stiffness can change rapidly with element rotation, we need a rotational event to control element rotation. Because stiffness changes continuously with element rotation, there are no natural rotational events. Therefore, we artificially define a discrete rotational event, namely an incremental angle. If the element rotates more than say ten degrees since the last reformulation of stiffness, element stiffness has changed significantly. We must then incrementally update the element stiffness to prevent too large a change in rotational (geometric) stiffness in a given step or substep.

We formulate the cable element to remain defined and non-singular even for the most extreme deformations or rotations. For example, the finite cable element is undefined when its two nodes coincide. To avoid this singularity, we simply re-position one node to a slightly different location when both nodes have equal coordinates. In this case, the numerical precision of the computer determines the exact meaning of equal.

3.4 Numerical Cable Super-Element

With a stable finite cable element, we now develop a reliable cable super-element. The heart of this cable super-element is a submodel of serially-connected finite cable elements. A condensation procedure converts the cable submodel into a super-element. This dynamic cable super-element is much more capable and versatile than the static semi-analytical one described in Section 3.2.

3.4.1 Cable Submodel. Each cable submodel consists of several serially-connected finite cable elements. The required number of elements depends on the local loading conditions and the local curvature of the catenary. We designate nodes internal to the submodel as local nodes because they connect finite cable elements and thus represent local submodel response. We designate the two nodes at the ends of the submodel as global nodes because they represent connections to other super-elements and thus represent global model response.

For the finite cable element and the cable submodel, we choose to associate all nodes (local and global) with the global coordinate system. We do this because the global direction of gravity is important for response of both the local finite cable element and the global cable super-element. For other types of submodels, we are free to associate local nodes with any coordinate system.

Using the traditional direct stiffness method of structural analysis, we assemble all the serially connected finite cable elements into a submodel (Row and Powell, 1978) as shown in Figure 3.8.

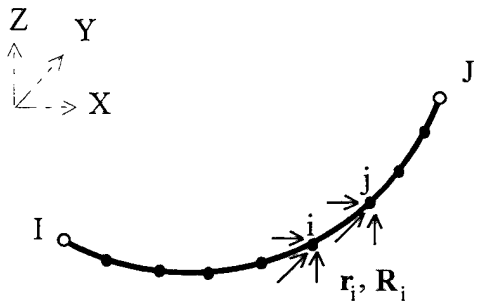


Figure 3.8. Assembly of finite cable elements into submodel.

Where:

I, J = global nodes

i, j = local nodes for element i

$\mathbf{r}_i, \mathbf{R}_i$ = displacement, force for nodes of element i

All global properties of the cable super-element derive from local solution of the cable submodel. The assembled matrices for the submodel (e.g., the Jacobian) show the same serial connectivity. All non-zero terms in the matrices are in a very narrow band along the matrix diagonal.

We prefer to base damping forces on real physical phenomena that occur at the element level. However, we know little about real damping principles at the element level. Therefore, Rayleigh damping theory (Clough and Penzien, 1975) provides a traditional option for specifying a mass and stiffness proportional type of damping.

$$\mathbf{C} = \alpha \mathbf{K} + \beta \mathbf{M} \tag{3.16}$$

Where:

α = factor of stiffness

β = factor of mass

$\mathbf{K}$ = submodel tangent stiffness

$\mathbf{M}$ = submodel mass

Traditional Rayleigh damping based on a constant initial structural mass and stiffness is not appropriate for a compliant structure. Time-varying damping is essential for modeling any large displacement, large rotation, structural problem. Damping must change as the structure changes. The time-varying form of the Rayleigh damping matrix changes in unison with the changing stiffness and mass matrices. In principle, this provides a damping force that is consistent with even the most radical nonlinear changes in structural shape or configuration.

3.4.2 Condensation to Super-Element. After obtaining a converged local solution, we condense the submodel to a super-element. The super-element has a reduced set of nodes external to the submodel as shown in Figure 3.9.

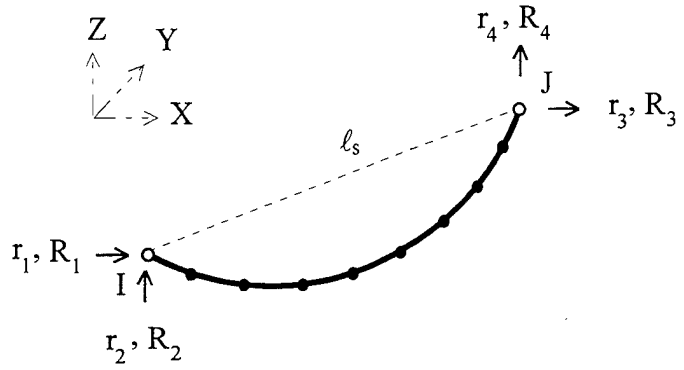


Figure 3.9. Numerical cable super-element.

Where:

I, J = global nodes (nodes of super - element)

ℓ_s = super - element (chord) length

$\mathbf{r}, \mathbf{R}$ = displacement, force for nodes of super - element

3.4.2.1 Traditional Condensation Procedure. For generating a static super-element, we can use the traditional static condensation procedure (Prezemienieki, 1963) for converting a submodel to a super-element. By condensing out nodes internal to each submodel (local nodes), this procedure creates smaller super-element arrays for direct assembly into the global stiffness matrix and static load vector.

For generating a dynamic super-element, we can extend this traditional procedure by operating on the Jacobian (effective stiffness) instead of simply the static stiffness. A dynamic condensation procedure formulates a full Jacobian matrix and related dynamic force vector for the super-element. This dynamic condensation procedure allows for step-linear mass and damping as well as the normal step-linear stiffness in the super-element.

The traditional condensation procedure starts by rearranging the assembled submodel arrays into internal (local) and external (global) partitions, whereby subscript L refers to local nodes and G refers to global nodes.

$$\begin{bmatrix} \mathbf{K}_{GG} & \mathbf{K}_{GL} \\ \mathbf{K}_{LG} & \mathbf{K}_{LL} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{r}_G \\ \Delta \mathbf{r}_L \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{R}_{GG} \\ \Delta \mathbf{R}_{LL} \end{bmatrix} \tag{3.17}$$

By algebraic manipulation, we eliminate the local degrees of freedom resulting in the following condensed incremental equations of motion for the super-element.

$$\mathbf{K}^* \, \Delta \mathbf{r} = \Delta \mathbf{R} \tag{3.18}$$

We form the specific Schur complements as follows:

$$\mathbf{K}^* = \mathbf{K}_{GG} - \mathbf{K}_{GL}\mathbf{K}_{LL}^{-1}\mathbf{K}_{LG} = \text{super - element Jacobian}$$
$$\Delta\mathbf{R} = \Delta\mathbf{R}_{GG} - \mathbf{K}_{GL}\mathbf{K}_{LL}^{-1}\Delta\mathbf{R}_{LL} = \text{incremental force for nodes of super - element}$$
$$\Delta\mathbf{r} = \Delta\mathbf{r}_G = \text{incremental displacement for nodes of super - element}$$

We obtain the incremental displacements at local nodes from incremental displacements at the global nodes.

$$\Delta\mathbf{r}_L = \mathbf{K}_{LL}^{-1}\left(\Delta\mathbf{R}_{LL} - \mathbf{K}_{LG}\Delta\mathbf{r}_G\right) \tag{3.19}$$

3.4.2.2 Special Condensation Procedure. The traditional condensation method requires that we re-order the end nodes of the submodel. This destroys the special serial connectivity of the cable submodel and the diagonal nature of the Jacobian matrix.

To avoid re-ordering, we develop a special condensation method for converting the cable submodel to a super-element. Instead of rearranging terms, we partition the assembled submodel Jacobian into three. Subscripts "I" and "J" refer to the two global nodes of the super-element and subscript "L" refers to the local nodes.

$$\begin{bmatrix} \mathbf{K}_{II} & \mathbf{K}_{IL} & \\ \mathbf{K}_{LI} & \mathbf{K}_{LL} & \mathbf{K}_{LJ} \\ & \mathbf{K}_{JL} & \mathbf{K}_{JJ} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{r}_I \\ \Delta\mathbf{r}_L \\ \Delta\mathbf{r}_J \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{R}_{II} \\ \Delta\mathbf{R}_{LL} \\ \Delta\mathbf{R}_{JJ} \end{bmatrix} \tag{3.20}$$

We pre-multiply both sides of equation (3.20) by the following matrix:

$$\begin{bmatrix} \mathbf{I} & -\left(\mathbf{K}_{IL}\mathbf{K}_{LL}^{-1}\right) & \\ & \mathbf{I} & \\ & -\left(\mathbf{K}_{JL}\mathbf{K}_{LL}^{-1}\right) & \mathbf{I} \end{bmatrix} \tag{3.21}$$

Then we eliminate the local nodes, resulting in the following super-element representation.

$$\mathbf{K}^*\,\Delta\mathbf{r} = \Delta\mathbf{R} \tag{3.22a}$$

Where:

$$\mathbf{K}^* = \begin{bmatrix} \mathbf{K}_{II}^* & \mathbf{K}_{IJ}^* \\ \mathbf{K}_{JI}^* & \mathbf{K}_{JJ}^* \end{bmatrix} = \text{super - element Jacobian}$$

$$\Delta\mathbf{R} = \begin{bmatrix} \Delta\mathbf{R}_{II}^* \\ \Delta\mathbf{R}_{JJ}^* \end{bmatrix} = \text{increment force for nodes of super - element}$$

$$\Delta \mathbf{r} = \begin{bmatrix} \Delta \mathbf{r}_I \\ \Delta \mathbf{r}_J \end{bmatrix} = \text{incremental displacement for nodes of super - element}$$

We form the specific Schur complements as follows:

$$\begin{aligned}
\mathbf{K}_{II}^{*} &= \mathbf{K}_{II} - \mathbf{K}_{IL}\mathbf{K}_{LL}^{-1}\mathbf{K}_{LI} \\
\mathbf{K}_{JJ}^{*} &= \mathbf{K}_{JJ} - \mathbf{K}_{JL}\mathbf{K}_{LL}^{-1}\mathbf{K}_{LJ} \\
\mathbf{K}_{IJ}^{*} &= -\mathbf{K}_{IL}\mathbf{K}_{LL}^{-1}\mathbf{K}_{LJ} \\
\mathbf{K}_{JI}^{*} &= -\mathbf{K}_{JL}\mathbf{K}_{LL}^{-1}\mathbf{K}_{LI} \\
\Delta\mathbf{R}_{I}^{*} &= \Delta\mathbf{R}_{II} - \mathbf{K}_{IL}\mathbf{K}_{LL}^{-1}\Delta\mathbf{R}_{LL} \\
\Delta\mathbf{R}_{J}^{*} &= \Delta\mathbf{R}_{JJ} - \mathbf{K}_{JL}\mathbf{K}_{LL}^{-1}\Delta\mathbf{R}_{LL}
\end{aligned} \tag{3.22b}$$

We obtain the local nodal displacements of the submodel from the super-element displacements by a recovery method.

$$\Delta\mathbf{r}_L = \mathbf{K}_{LL}^{-1}\left(\Delta\mathbf{R}_{LL} - \mathbf{K}_{LI}\Delta\mathbf{r}_I - \mathbf{K}_{LJ}\Delta\mathbf{r}_J\right) \tag{3.23}$$

3.4.3 Super-Element Events. In each local/global cycle, it is often not necessary to update every submodel. For example, if a particular submodel is behaving quite linearly, then a constant super-element Jacobian for this submodel is sufficient for the global model. Super-element events are a fast way to indicate how often to update the state and solve the linearized equations of any particular submodel. We choose the following four events for the cable super-element:

- Change in state from slack to taut
- Change in state from taut to slack
- Incremental rotation in excess of a given rotational tolerance
- Significant change in load on the super-element

To compute accurate super-element events, we must use local information. However, we can approximate super-element events using only global information. For example, the slack/taut length for the super-element is equal to the summation of globally known slack/taut lengths for all the elements that make up the super-element. The super-element is taut or slack, when the distance between its global nodes is greater or less than this super-element length, respectively. Obviously we can pose a dynamic condition, whereby the super-element will be taut when not all of the elements within the submodel are taut. Hence, the super-element slack/taut length is only an approximate state variable.

Since the loads and stiffness of a super-element are rotationally dependent, we also desire an event for indicating when the super-element axis exceeds a certain increment of rotation. We recommend an angle of ten degrees for this rotational threshold.

Unlike the simple finite cable element, the cable super-element has an internal distribution of loads that is important to the solution of the cable submodel. Since we already

have a rotational event that will indicate changes in load orientation, we need a global event that indicates significant changes in the magnitude of the equivalent super-element load.

3.5 Floating Buoys

Often in a compliant marine structure, there are buoy substructures that float on or below the surface of the water. If the buoy remains submerged, a force vector may suffice for its static representation. If the buoy crosses the water surface or if it has significant dynamic behavior, then a buoy element is necessary. If the buoy has a geometrically complex shape with structural flexibility, then a full buoy super-element may be necessary. We develop a simple buoy element and show the development path for a more complex super-element.

3.5.1 Simple Buoy Element. The buoy element has one node, with only three translational degrees of freedom, as shown in Figure 3.10. With only one node, the buoy element represents only simple hydrostatic stiffness and resistance normal to the fluid surface plus elemental hydrodynamic forces.

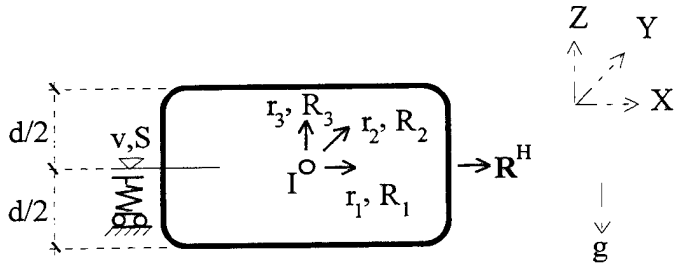


Figure 3.10 Simple buoy element.

Where:

d = depth of buoy

v, S = element deformation and action

$\mathbf{r}, \mathbf{R}$ = displacement, force for node of element

$\mathbf{R}^H = \mathbf{R}^M + \mathbf{R}^C$ = hydrodynamic force (see Section 2.2.2)

I = element node

Figure 3.11 shows the action-deformation relationship for the buoy element. Element deformation is equal to the vertical component of nodal displacement. If the water plane area remains essentially constant with vertical motion of the buoy, the action-deformation relationship is constant (i.e., linear stiffness).

If the buoy sinks below or lifts above the fluid surface, the action-deformation relationship is discontinuous (i.e., multi-linear stiffness) as shown in Figure 3.11. An element event is excellent for handling these discontinuities in vertical stiffness. Horizontal stiffness is always zero.

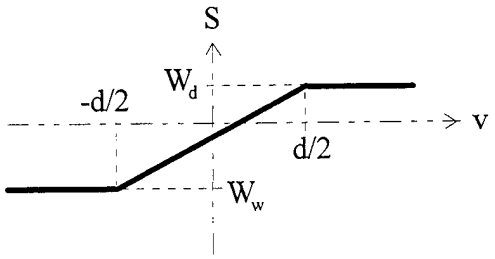


Figure 3.11 Action-deformation relation -- simple buoy element.

Where:

W_d = dry weight of buoy
W_w = wet (submerged) weight of buoy

If the water plane area changes with vertical motion of the buoy, the action-deformation relationship is nonlinear (i.e., variable stiffness).

The simplest of kinematic compatibility and force equilibrium relationships apply to the simple buoy element.

$$v = r_3 \qquad (3.24)$$
$$R_3 = S$$

3.5.2 Complex Buoy Element or Super-Element. The physical interaction between the buoy and the water surface may be more complex than what a simple buoy element with three translational degrees of freedom can represent. Rotational resistance with rotational degrees of freedom may be important for the desired response of the buoy. This is particularly true if there are several different attachment points that connect to other substructures.

Shown in Figure 3.12, a typical semisubmersible buoy element would have one major node (I) and three minor nodes (J, K, L). The major node would have 6 degrees of freedom and be directly involved in solution of the element response. The minor nodes would represent connections between the semisubmersible buoy and its mooring cables substructures. We would slave the response of the minor nodes to the major node (I) by means of rigid transformations (Cook, 1989). For large rotations of the semisubmersible buoy, these transformations are nonlinear.

Rotational degrees of freedom would allow for relative motion hydrodynamic force computation on each individual member of the semisubmersible (Paulling, 1977). We would make this hydrodynamic computation on each wet differential piece of each hydrodynamic element that makes up the rigid semisubmersible buoy. This one-way hydrodynamic load calculation would depend on the instantaneous relative motion between each differential piece of the buoy and surrounding water particles.

A full super-element model of the semisubmersible buoy would require local nodes spread throughout the semisubmersible substructure. Not only would these nodes monitor the structural displacements of the flexing hull but they would also monitor the relative motion

between the fluid field and each member of the structure. Real fluids with turbulence effects develop a temporal and spatial interaction between the flowing fluid and the structural members.

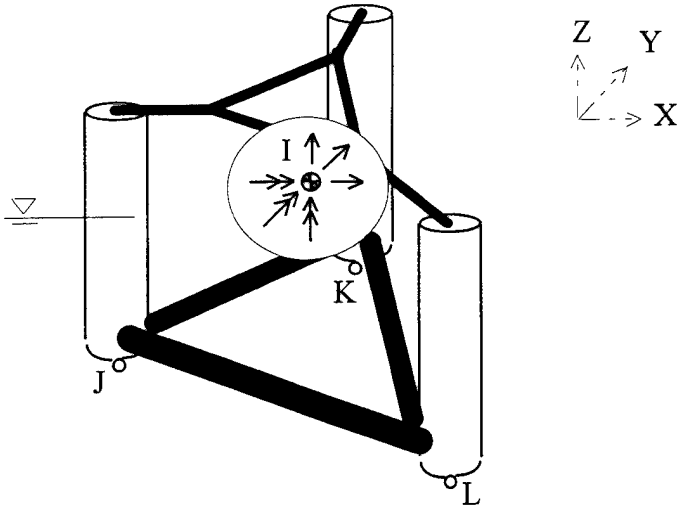


Figure 3.12 Complex semisubmersible buoy super-element.

A local/global model with a semisubmersible buoy super-element would keep track of all this information and keep it private to the corresponding semisubmersible buoy submodel.

3.6 Seafloor Plow

Towing a plow along the seafloor is often the preferred method for burying cables, recovering seafloor minerals, or measuring seafloor soil properties. The performance of this plow depends on careful design and analysis using a seafloor/plow interaction submodel. A simple seafloor plow element is adequate for determining responses of the top of the tow cable far from the seafloor plow. A complex plow super-element is better for simulating the actual response of the plow substructure. We develop a simple seafloor plow element and show the development path for a more complex super-element.

3.6.1 Simple Seafloor Plow Element. A seafloor plow has a blade that penetrates the seafloor and a sled that rides on the seafloor surface. The seafloor plow element has one node, with only two (translational) degrees of freedom, as shown in Figure 3.13. With only one node, the seafloor plow element represents simple soil resistance longitudinal and normal to the seafloor. For this simple element, we assume that all forces act through the single node (I). We also assume that the cable substructure for towing the plow element connects at node (I).

The seafloor may change elevation and slope. Consequently, the normal vector to the seafloor under the plow sets the local normal direction of the plow element. We fix the plow in the local transverse direction, since the blade does not permit any significant transverse motion. The local orientation of the attached tow cable sets the local longitudinal direction of the plow

element. We transform local element displacements directly from the global displacements. We base this transformation on directional
cosines of the angles between the local and global axes.

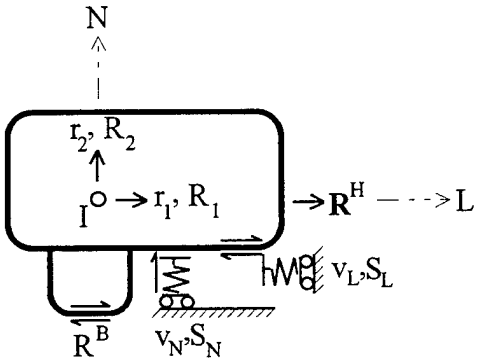


Figure 3.13 Simple seafloor plow element.

Where:

v, S = deformation and action of seafloor plow sled

$\mathbf{r}, \mathbf{R}$ = displacement, force for nodes of element

$\mathbf{R}^H$ = fluid force (see Section 2.2.2)

R^B = viscous resistance force at the blade

L, N = local coordinates (longitudinal, normal to the seafloor)

I = element node

The viscous resistance force of the blade, which acts only in the local longitudinal direction of the plow element, is as follows:

$$R^B = \frac{\alpha}{\beta} \tag{3.25}$$

Where:

$$\alpha = \alpha_0\left(1 + \upsilon|\dot{v}_L|\right)$$

$$\beta = \sqrt{1 + (\alpha / \chi)^2}$$

α_0, υ, χ = proprietary constants that depend on specific soil type and plow shape

Figure 3.14 shows the action-deformation relationships in the normal and longitudinal directions for the sled of the seafloor plow. Element events are excellent for handling the discontinuities in the stiffness of these action-deformation relationships.

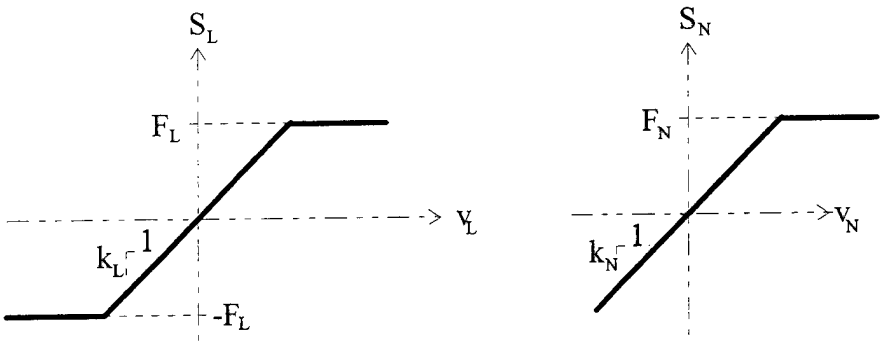


Figure 3.14 Action-deformation relationships- simple seafloor plow.

Where:

F = force limit

k = stiffness

Normal deformation represents local displacement normal to a seafloor elevation reference. A table of seafloor elevations versus position on the seafloor provides the necessary changing ground reference. The normal force limit is wet weight of the seafloor plow. Beyond this force limit the seafloor plow lifts its sled off the seafloor. Otherwise, the sled deforms the seafloor in a normal direction to the seafloor.

The longitudinal force limit represents frictional slippage between the seafloor and the sled. Below this limit, the sled deforms soil in a longitudinal direction to the seafloor. Above this limit, the sled slips relative to the soil. Longitudinal deformation represents local displacement in the instantaneous direction of plowing from a reference "no-slip" condition. We redefine longitudinal deformation at each step that the sled slips.

Coulomb frictional theory (Argyris and Mlejnek, 1991) gives the instantaneous longitudinal force limit.

$$F_L = \mu F_N = \text{limiting frictional force along contact surface} \qquad (3.26)$$

Where:

μ = frictional factor (assumed = 0.3)

F_N = static force normal to contact surface

In addition to element events for handling discontinuities in the action-deformation relationships, we also define an event for each discrete change in seafloor slope. A change in seafloor slope can cause a change in stiffness components. It is necessary to correct for the new orientation of the seafloor at the exact change in seafloor slope. Otherwise, we will generate very large seafloor actions and thus very large force imbalances.

Simple kinematic compatibility and force equilibrium relationships apply to the simple seafloor plow element.

$$dv = a\, dr_3$$
$$dR_3 = a^T dS \qquad (3.27)$$

Where:

a = directional cosine of angles between local and global axes

3.6.2 Complex Seafloor Plow Element or Super-Element. The physical interaction between the plow and the seafloor may be more complex than what a simple one-node element with two degrees of freedom can represent. No rotational resistance, no spatially variable soil pressures, no soil deformation, no complex soil damping, and no transverse responses are key limitations of the simple one-node submodel. Expanding the seafloor plow submodel into a larger submodel with more degrees of freedom is necessary to remove these limitations.

Shown in Figure 3.15, a more complex seafloor plow element would have one major node (I) and a minor node (J). The major node would have 6 degrees of freedom and be directly involved in solution of the element response. The minor node would represent an eccentric connection to a tow cable substructure. We would slave the response of the minor node (J) to the major node (I) by rigid transformation (Cook, 1989). For large rotations of the plow element, these transformations are nonlinear.

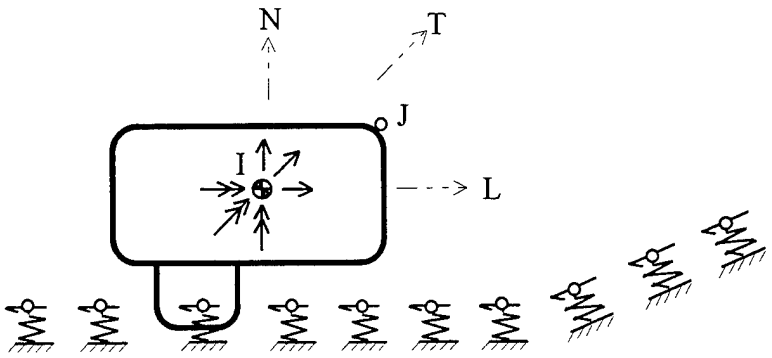


Figure 3.15 Complex seafloor plow element or super-element.

The rotational degrees of freedom would allow for relative-motion geotechnical force computation on each individual member of the seafloor plow (similar to the PSAS element in PMB, 1989). The bed of springs (soil subelements) along the seafloor would represent this soil/structure interaction computation. We would require a minor node at each soil subelement to monitor soil deformation with seafloor location as shown in Figure 3.15. These minor nodes would monitor the local interaction between the soil and the sled and between the soil and the blade. As the sled or blade makes contact with each soil subelement, the minor node of the subelement becomes a part of the response calculation for the seafloor plow super-element.

Real soil plowing with plastic deformation and fluid porosity effects requires both temporal and spatial interactions between the geotechnical degrees of freedom and the plow

structural members (Taylor, 1982). The resistance of the soil, both normal and longitudinal to the surfaces of the seafloor plow depend on the instantaneous relative motion between each differential piece of the plow and surrounding soil particles.

A local/global model with a seafloor plow super-element would keep track of all this information and keep it private to the corresponding seafloor plow submodel.

4. IMPROVED NONLINEAR SOLUTION STRATEGIES

A local/global computational framework provides a balance between local and global nonlinear solution requirements. Although the framework allows for different nonlinear solution strategies in each local submodel, there must be sufficient commonality between the local solution strategies to guarantee effective global integration.

Section 4.1 introduces this common nonlinear solution strategy. Section 4.2 describes how we specialize this strategy for robust local solution of cable submodels. Section 4.3 highlights specific techniques that assure robust and reliable nonlinear structural simulation both at the local and global domains.

4.1 Nonlinear Solution Strategy With Event Control

Nonlinear step-by-step integration of a progressively changing system of incremental equations is the preferred computational method for simulating the structural response of compliant structures. Compliant marine structures, particularly ones dominated by cable substructures, derive their stiffness, and thus their ability to support load, from shape and internal force. There are two main difficulties with using traditional Newton-based incremental/iterative strategies for nonlinear simulation of compliant marine structures (Webster, 1980):

- Traditional strategies require a well posed starting estimate. In other words, the initial shape must be in fair equilibrium with forces. If not, the structural system is often ill-conditioned, and the solution often diverges.
- Traditional Newton-based procedures have a limited interval of convergence around the correct solution. If ill-conditioning develops along the solution path, then the solution often diverges.

Ill-conditioning is an indication that the nonlinear strategy has reached a turning point in its attempt to follow the correct nonlinear solution path. It is essential that we give special treatment to these turning points. In chapter three, we identified these turning points as element events and developed new elements that gave clear physical meaning to these events. Now we will use the numerical control provided by these events to make a revolutionary improvement in our nonlinear solution strategies.

We base our improved nonlinear solution strategies on two previously discussed ingredients:

- The trapezoidal rule for step-by-step time integration
- The Newton-Raphson procedure for nonlinear stepping

We modify both of these classical ingredients to include event control in a way that achieves remarkable solution robustness (speed, efficiency, and stability) in the nonlinear simulation of compliant marine structures. The essential goal of a nonlinear step-by-step solution is to determine corresponding increments of displacement and load for which all forces are in acceptable equilibrium.

For our improved solution strategies, each step can be any desired size and can be static or dynamic. In a static step, we seek the increment of displacement that corresponds to a given increment of static load. In a dynamic step, we seek the increment of displacement that corresponds to a given increment of time. Naturally, an increment of dynamic load corresponds to any given increment of time. This load or external force can change with displacement of the structure.

The main difference between a dynamic and a static step is that the dynamic step includes inertial and viscous forces. Otherwise, the solution strategies for both the static and dynamic steps are essentially the same. At each step of the simulation, there may be one or more substep solutions for reducing force unbalance in the structural system. We use a Jacobian to "point" the solution in the right direction at each substep. The Jacobian may change at each substep. The Jacobian is the tangent stiffness or the effective tangent stiffness for a static or a dynamic step, respectively.

Depending on how small we choose the elements, the static numerical solution closely approximates the continuous spatial configuration of the actual structure. Also, depending on how small we choose the time steps, the dynamic step-by-step numerical simulation closely approximates the continuous temporal response of the actual structure.

4.1.1 Modified Trapezoidal Rule. Solution of a dynamic step starts by assuming a specific finite difference relationship between nodal acceleration, velocity, and displacement for any increment of time. For compliant nonlinear structures, we prefer to use the trapezoidal rule as this finite difference (Belytschko and Hughes, 1993). The trapezoidal rule is a simple one-step implicit Newmark method. Its numerical stability is unconditional of time step size; and it produces no numerical damping. The trapezoid rule has remarkable stability for nonlinear problems (Dahlquist, 1963).

The trapezoidal rule assumes that acceleration is a constant average of the values of acceleration at the beginning and end of any given time step, as shown in Figure 4.1. By integrating over the step, velocity varies linearly and displacement varies as a quadratic over the given time step. The trapezoidal rule implies that load varies linearly over the given time step.

The trapezoidal rule possesses simple equations for velocity and displacement in terms of the known and the unknown acceleration at a given time step.

$$\dot{\mathbf{r}}_m = \dot{\mathbf{r}}_{m-1} + \frac{\Delta t_m}{2}\left(\ddot{\mathbf{r}}_{m-1} + \ddot{\mathbf{r}}_m\right)$$
$$\mathbf{r}_m = \mathbf{r}_{m-1} + \frac{\Delta t_m^2}{4}\left(\ddot{\mathbf{r}}_{m-1} + \ddot{\mathbf{r}}_m\right) \qquad (4.1)$$

Where:

m = 1,2,3,..= step number (m = 0 at beginning of simulation)

Δt = time step (may vary arbitrarily with m)

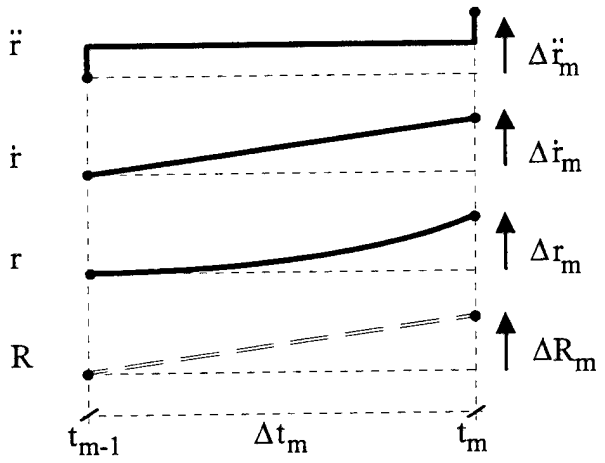


Figure 4.1 Trapezoidal rule for time integration

Rearranging terms so that incremental displacement becomes the unknown, we get the following two equations for step increments of velocity and acceleration.

$$\begin{aligned}\Delta\dot{\mathbf{r}}_m &= 2\Delta d_m \Delta\mathbf{r}_m - 2\dot{\mathbf{r}}_{m-1} \\ \Delta\ddot{\mathbf{r}}_m &= 2\Delta d_m \Delta\dot{\mathbf{r}}_m - 2\ddot{\mathbf{r}}_{m-1}\end{aligned} \tag{4.2}$$

With our chosen nonlinear solution strategy, events may restrain the solution at any substep. The equations for incremental velocity and acceleration require modification as shown below to allow event-based substep iteration.

$$\begin{aligned}\Delta d_m &= {}^{1}\!/_{\Delta t_m} \left(\equiv 0 \text{ for a static step}\right) \\ \Delta\mathbf{r}'_n &\Leftarrow \left(f_{\min}\right)_n \Delta\mathbf{r}_n \\ \Delta\dot{\mathbf{r}}_n &= 2\Delta d_m \left(f_{\min}\right)_n \Delta\mathbf{r}'_n - 2\left(f_{\min}\right)_n \left(f_{\text{rem}}\right)_n \dot{\mathbf{r}}_o \\ \Delta\ddot{\mathbf{r}}_n &= 2\Delta d_m \Delta\dot{\mathbf{r}}_n - 2\left(f_{\min}\right)_n \left(f_{\text{rem}}\right)_n \ddot{\mathbf{r}}_o\end{aligned} \tag{4.3}$$

Where:

n = 1,2,3,..= substep number (n = 0 at beginning of step)

$\left(f_{\min}\right)_n = min\left\{\left(f_i\right)_n\right\}$ = minimum governing event factor for all i

$(f_i)_n$ = minimum event factor for element i

$(f_{rem})_n = (f_{rem})_{n-1}\left(1-(f_{min})_n\right)$ = remaining load factor

$(f_{rem})_0 \equiv 1$

The following limits to the range of values apply:

$$0 \le f_{min} \le 1$$
$$0 \le f_{rem} \le 1 \tag{4.4}$$

The remaining load factor, which decreases from one to zero over any given step, tracks how much of the response in that step still needs computing. Summation of the coefficients of equation (4.3) over any given step with any arbitrary set of events will produce the normal expressions for the trapezoidal rule, equation (4.2). In other words, the following summation holds.

$$\sum_{n=1}^{\infty}(f_{min})_n(f_{rem})_n = 1 \tag{4.5}$$

By substituting the expressions for incremental displacement, velocity, and acceleration into equation (2.2) and rewriting in residual form, we get the following set of algebraic equations. These equations are for incremental representation of both the static and dynamic structural systems.

$$\mathbf{K}_n^* \Delta\mathbf{r}_n = \mathbf{R}_n^U \tag{4.6}$$

Where:

$\mathbf{K}_n^* = 4\Delta d_m{}^2\mathbf{M}_m + 2\Delta d_m\mathbf{C}_m + \mathbf{K}_n$ = Jacobian (effective tangent stiffness)

$\mathbf{R}_n^U = \mathbf{R}_n^E - \mathbf{R}_n^I + f_{rem}\mathbf{R}_m^D$ = unbalanced force

$\mathbf{R}_n^E$ = externally applied force (configuration - dependent load)

$\mathbf{R}_n^I$ = internal resisting force = $\mathbf{R}_n^M + \mathbf{R}_n^V + \mathbf{R}_n^S$

$\mathbf{R}_n^M \equiv \mathbf{M}_m\ddot{\mathbf{r}}_{n-1}$ = inertial resisting force (equal to 0 for static step)

$\mathbf{R}_n^V \equiv \mathbf{C}_m\dot{\mathbf{r}}_{n-1}$ = viscous resisting force (equal to 0 for static step)

$\mathbf{R}_n^S$ = static resisting force (assembled from **S** of each element)

$\mathbf{R}_m^D = (2\ddot{\mathbf{r}}_0 + 4\Delta d_m\dot{\mathbf{r}}_0)\mathbf{M}_m + (2\dot{\mathbf{r}}_0)\mathbf{C}_m$ = constant force of time integration

Equation (4.6) is similar to the traditional equations of motion (2.18), except for a few notable changes. We use the trapezoidal rule as the finite difference, and we include the constant force of time integration directly in the unbalance force calculation. We compute a new constant

force of time integration at the beginning of each time step. As such, the constant force of time integration participates directly in determining convergence of the step-by-step solution.

Using the trapezoidal rule, we compute velocity and acceleration from corresponding increments of displacement. It is important to note that values for displacement, velocity, and acceleration (and thus time) are realistic only if force equilibrium exists. For example, the Newton-Raphson method enforces force equilibrium only at the end of a step. Therefore, values for displacement, velocity, and acceleration are realistic only at the end of the time step; substep values are unrealistic.

4.1.2 Newton-Raphson With Event Control. The purpose of event control is to help follow the true solution path as closely as possible. In this way, the error in the resulting force equilibrium remains small, and the unbalanced force across the structure never gets too large. Without proper control, the unbalanced force may become so large that a subsequent Jacobian may not have a chance to bring the solution back to force equilibrium.

Event control works specifically by identifying "events" that indicate significant nonlinear changes in the equations of motion at the element level. As such, an event is an element-based indicator that the solution has reached a turning point and the Jacobian needs updating. We prefer to use physically meaningful events, since these naturally give the best chance of approaching real physical stability. Events need not represent real discrete changes in the system. Alternatively, they can represent real continuous changes that have been arbitrarily discretized.

Instead of advancing a full Newton solution step, we restrain the displacement increment to where the first event occurs along the projected solution path. The restraint consists of factoring the computed increment of displacement to a magnitude that is just past the event. Since the system is linearized at each substep, this factoring of displacement has the same effect as factoring the increment of load.

Figure 4.2 depicts the chosen Newton-Raphson nonlinear solution strategy with event control. For graphic simplicity in the figure, we show only one constant load step and only one degree of freedom. Also for graphic simplicity, we show only three substeps for iterative convergence to the final solution.

Two specific events e_1 and e_2 delineate the three substeps. In Figure 4.2, the solution path is a light solid line and the true equilibrium path is a heavy solid line. For each event, the dashed line would represent the traditional Newton solution step, and the dash-dot line represents the restraint imposed by the corresponding event. In order to follow the true equilibrium path as closely as possible from point a to point b, we update the Jacobian at each substep.

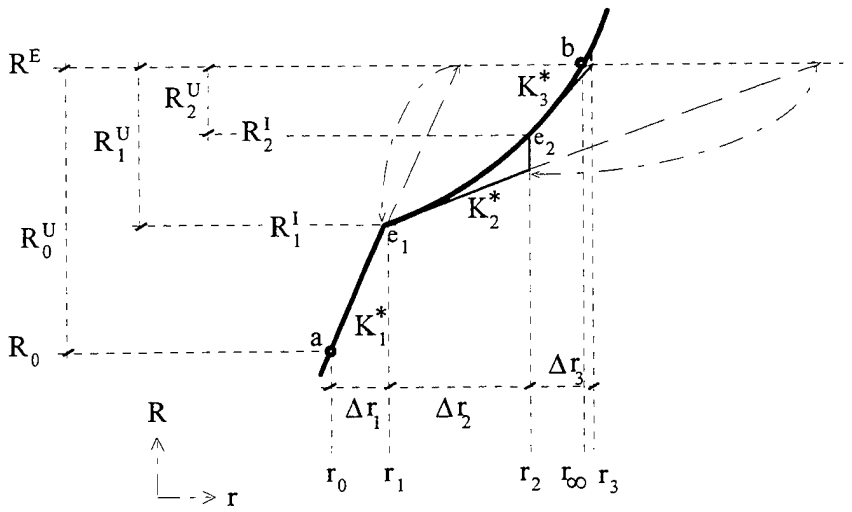


Figure 4.2 Newton-Raphson step with event control.

Where:

$n = 1,2,3,...$= substep number (0 = start of step)

$\mathbf{R}_n$ = total load

$\mathbf{r}_n$ = total displacement

$\mathbf{R}_n^U$ = unbalanced force

$\mathbf{R}_n^I$ = internal resisting force

$\mathbf{K}_n^*$ = Jacobian (effective tangent stiffness)

$\Delta\mathbf{r}_n$ = increment of displacement

$\mathbf{R}^E$ = total load for step

$\mathbf{R}^E = \mathbf{R}_n \quad \text{as } n \rightarrow \infty$

e_n = element event

If the event represents a true discrete event (e.g., axial event e_1), the numerical solution follows the true equilibrium path. However, if the event is arbitrary (e.g., rotational event e_2), the numerical solution only approximately follows the true path.

For a multi-element problem, we must calculate an event factor for each element and then select the minimum factor among all the elements. This minimum factor governs the scaling of the load increment for that substep. We repeat for each successive substep until we have exhausted all possible events, have applied the full load for the step, and have reached the desired level of force equilibrium.

Figure 4.3 shows the flow of information through the required computational modules for successive steps. It is beneficial to contrast this flow of information with that for the traditional Newton-Raphson solution strategy, Figure 2.3. The traditional strategy has two different substep loops, one each for the correcting and advancing phases.

With configuration-dependent loads, the load can change substantially at every substep. Consequently, every substep has both an advancing and correcting nature. Event control primarily determines the size of a substep. If there is no event, only force imbalance determines if another substep follows the current one. Both event control and force imbalance determine the total number of substeps in any given step. To assure robust nonlinear solution simulation, we need no user intervention such as changing the step size or applying the load in several steps.

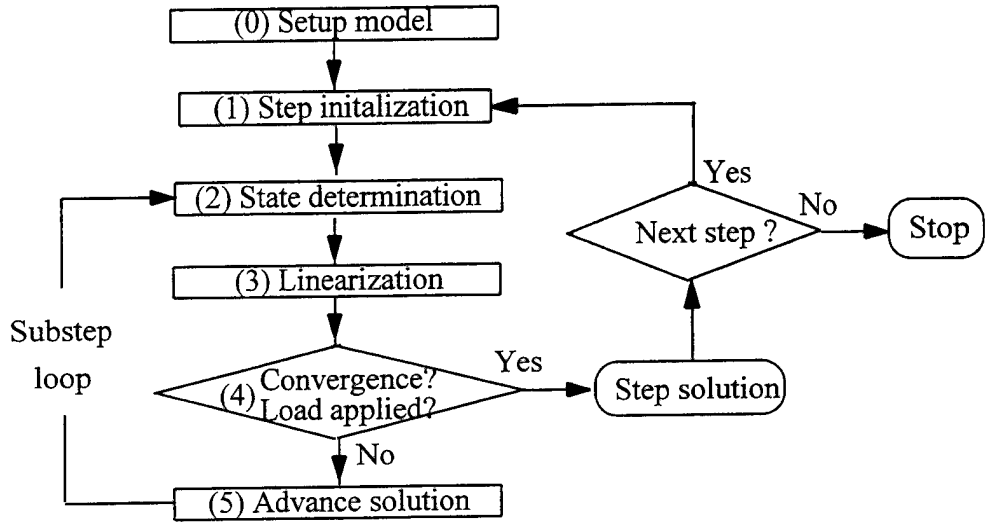


Figure 4.3 Newton-Raphson solution strategy with event control.

The list below describes each of the computational modules in Figure 4.3.

(0) Setup model:

Setup initial nodal geometry, define initial structural model, and set all fixed and variable properties for each element.

$$\begin{aligned} n &= \infty \quad (\text{last substep in step}) \\ \mathbf{r}_\infty &= \text{given} \\ \dot{\mathbf{r}}_\infty, \ddot{\mathbf{r}}_\infty &= \mathbf{0} \\ \mathbf{R}_\infty &= \mathbf{0} \quad (\text{or given}) \\ \mathbf{R}_\infty^U &= \mathbf{0} \end{aligned} \tag{4.7}$$

(1) Step initialization:

Initialize step, reset event factors, and redefine model. We redefine the model by modifying the variable properties for each element as needed. Also, we add or remove

nodes or elements to model as needed. For example, to pay out a length of cable, we increase the slack/taut length of the last element by the desired pay-out length. See Section 4.2.4 for more detail on cable paying.

$$
\begin{aligned}
f &= 1 \\
\mathrm{n} &\Leftarrow 0 \\
\mathbf{r}_0, \dot{\mathbf{r}}_0, \ddot{\mathbf{r}}_0 &\Leftarrow \mathbf{r}_\infty, \dot{\mathbf{r}}_\infty, \ddot{\mathbf{r}}_\infty \\
\mathbf{R}_0 &\Leftarrow \mathbf{R}_\infty \\
\mathbf{R}_0^{\mathrm{U}} &\Leftarrow \mathbf{R}_\infty^{\mathrm{U}}
\end{aligned}
\tag{4.8}
$$

(2) State determination:

Determine state variables for each element. State variables are element data that change at every substep, such as element deformation and action. Determine total external force on each element.

$$
\begin{aligned}
\mathbf{S}, \mathbf{v} &= \mathfrak{I}\left(\mathbf{r}_{\mathrm{n}}, \ldots\right) \\
\mathbf{R}^{\mathrm{E}} &= \mathfrak{I}\left(\mathbf{R}_{\mathrm{n}}, \mathbf{r}_{\mathrm{n}}, \dot{\mathbf{r}}_{\mathrm{n}}, \ddot{\mathbf{r}}_{\mathrm{n}} \ldots\right) \\
\mathrm{n} &\Leftarrow \mathrm{n}+1
\end{aligned}
\tag{4.9}
$$

(3) Linearization:

Compute mass, damping, and tangent stiffness for all elements. Assemble Jacobian (effective stiffness matrix).

$$
\begin{aligned}
\mathbf{K} &= \mathfrak{I}\left(\mathbf{S}, \mathbf{v}, \ldots\right) \\
\mathbf{K}_{\mathrm{n}}^{*} &= \mathfrak{I}\left(\Delta \mathrm{t}, \mathbf{M}, \mathbf{C}, \mathbf{K}, \ldots\right) \text{ per equation (4.6)}
\end{aligned}
\tag{4.10}
$$

(4) Convergence? Load applied?

Calculate nodal force for each element and assemble into the internal resisting force vector. Compute damping and inertial resisting forces. For Rayleigh-type damping, this requires updated mass and damping matrices. Compute unbalanced force. If the unbalanced force has adequate convergence and if system has full load applied, skip to module (1) for next step.

$$
\mathbf{R}_{\mathrm{n}}^{\mathrm{U}} = \mathfrak{I}\left(\mathbf{R}^{\mathrm{E}}, \mathbf{M}, \mathbf{C}, \mathbf{S}, \dot{\mathbf{r}}_{\mathrm{n}-1}, \ddot{\mathbf{r}}_{\mathrm{n}-1}, \ldots\right) \text{ per equation (4.6)} \tag{4.11}
$$

(5) Advance solution:

Calculate a displacement increment by solving the following equation:

$$\mathbf{K}_n^* \Delta\mathbf{r}_n = \Delta\mathbf{R}_n \equiv \mathbf{R}_n^U \tag{4.12}$$

Compute minimum event factor based on smallest event factor of all elements.

$$f_{\min} = \mathfrak{I}\left(\mathbf{S}, \mathbf{v}, \mathbf{r}_{n-1}, \mathbf{r}_0, \Delta\mathbf{r}_n, \ldots\right), \quad 0 \le f_{\min} \ge 1 \tag{4.13}$$

Update the displacements, velocities, and accelerations. Update factors for tracking solution. Go to module (2) for next substep.

$$\mathbf{r}_n, \dot{\mathbf{r}}_n, \ddot{\mathbf{r}}_n = \mathfrak{I}\left(\Delta t, f_{\min}, \Delta\mathbf{r}_n, \mathbf{r}_{n-1}, \dot{\mathbf{r}}_{n-1}, \ddot{\mathbf{r}}_{n-1}, \dot{\mathbf{r}}_0, \ddot{\mathbf{r}}_0\right) \text{ per equation (4.3)} \tag{4.14}$$

4.2 Specialized for Local Solution

It is beneficial to specialize the chosen nonlinear solution strategy for each specific submodel. In this section, we specialize the chosen strategy for local solution of our cable submodel. Specialization for other types of submodels would follow a similar approach.

4.2.1 Cable Submodel Specialization. We choose to develop a specialized local nonlinear solution strategy that recognizes the high degree of geometric nonlinearity and serial connectivity of our cable submodel. We implement this local nonlinear solution strategy in the computer code MBDSIM as described in Chapter 5. This particular implementation allows the cable submodel to have nodal elements, which tie each node to a reference ground location.

4.2.1.1 List of Variables. The lists presented below defines the specific local and global variables for the cable submodel. All variables are for the current step, unless stated otherwise. The first list is for local variables representative of the submodel. The second list is for global variables representative of the super-element. Brackets [##] indicate the respective FORTRAN variable name used in the computer code MBDSIM.

Local variables:

t = simulation time (does not change during substep iterations)
Δt = time step
Δd = $1/\Delta t$ = inverse of time step (for static analysis, set equal to zero)
f = factor of displacement to which an event desires to restrain the solution
$f_{\min}$ = minimum event factor that governs for all elements
f_{apl} = applied factor of load step, whose response has been computed
f_{rem} = remaining factor of load step, whose response is still to be computed
$\mathbf{L}_0$ = taut/slack lengths of elements (a variable element property)
$\mathbf{r}$ = coordinates at cable ends
$\dot{\mathbf{r}}$ = velocity
$\ddot{\mathbf{r}}$ = acceleration

$\mathbf{r}_0$ = reference coordinate for nodal springs
$\dot{\mathbf{r}}_0$ = velocity at beginning of step
$\ddot{\mathbf{r}}_0$ = acceleration at beginning of step
$\mathbf{K}$ = tangent stiffness, including $\mathbf{K}^X$
$\mathbf{K}^X$ = tangent stiffness for nodal springs only
$\mathbf{C}$ = viscous damping
$\mathbf{M}$ = mass
$\mathbf{K}^*$ = Jacobian (effective stiffness)
$\mathbf{R}^E$ = force due to total element and field load
$\mathbf{R}^M$ = inertial resisting force
$\mathbf{R}^V$ = viscous resisting force
$\mathbf{R}^S$ = static resisting force
$\mathbf{R}^X$ = force due to deformation of nodal springs $= \mathbf{K}^X(\mathbf{r} - \mathbf{r}_0)$
$\mathbf{R}^D$ = constant force of time integration
$\mathbf{R}^U$ = unbalanced force
$\mathbf{R}_I^U$ = unbalanced force corresponding to the internal displacements only
η = maximum scalar norm of $\mathbf{R}^U$ or $\mathbf{R}_I^U$
$\Delta\mathbf{R}$ = force increment for equation solving
$\Delta\mathbf{r}$ = displacement increment due to $\Delta\mathbf{R}$
$\Delta\dot{\mathbf{r}}$ = velocity increment
$\Delta\ddot{\mathbf{r}}$ = acceleration increment

Global variables:

$\Delta\mathbf{q}$	=	specified displacement increment at two cable end nodes	[DDISE]
$\mathbf{Q}^S$	=	static super-element resisting force at cable end nodes	[RSTAT]
$\mathbf{Q}^V$	=	viscous super-element resisting force at cable end nodes	[RVISC]
$\mathbf{Q}^M$	=	inertial super-element resisting force at cable end nodes	[RINER]
$\mathbf{K}^Q$	=	condensed super-element stiffness for cable end nodes	[SGK]
$\mathbf{R}^Q$	=	condensed super-element load for cable end nodes	[SGR]

4.2.1.2 Algorithm for A Static or Dynamic Step. Below is the basic algorithm for solution of one static or one dynamic step of the submodel. A static step is simply a dynamic step with all dynamic terms in the equations of motion set equal to zero. We do this by simply setting the "inverse of time step" equal to zero. In the rare case that there is already adequate force balance upon entering the step, the applied load factor allows exit from the substep loop without making any substep solutions. Event control and load imbalance automatically choose all iterative substeps for the step. We generally prefer to compute a new Jacobian at every substep.

The selected starting configuration for each cable submodel is all nodes at rest and all element nodes in line between specified end positions of the cable. However, the solution strategy allows selection of just about any other reasonable starting configuration.

In the algorithm given below, brackets [##] indicate the respective FORTRAN subroutines name used in the computer code MBDSIM. The line numbers **(#)** correspond to the modules described in Figure 4.3.

Given from base program: $t, \Delta t, \Delta \mathbf{q}$

(0) Setup model.

If t = 0, initialize submodel:	[SETUP]
Initialize variables.	[INITIAL]
Input geometry and parameters.	[INPUT]
Generate nodes. Set all element properties.	[GENERATE]

(1) Step Initialization.

Redefine model, i.e., pay cable by changing $\mathbf{L}_0$.	[PAYOUT]
Initialize step:	[STEPINIT]
Set initial event factors, $f_{apl} = 1$, $f_{rem} = 1$.	
If dynamic step, set $\Delta d = 1/\Delta t$.	
If static step, set $\Delta d = 0$ and set $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}} = 0$.	
Copy $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$ to $\dot{\mathbf{r}}_0$, $\ddot{\mathbf{r}}_0$.	
Impose movements at cable ends $\mathbf{r}_0 = \mathbf{r}_0 + \Delta \mathbf{q}$.	

Substep loop (for a maximum number of substep cycles):

(2) State determination.

Compute $\mathbf{R}^S$, $\mathbf{R}^X$.	[RESPS]
Form $\mathbf{R}^E$ for each element.	[FORCE]

(3) Linearization.

Form $\mathbf{M}$ and $\mathbf{K}$ in tri-block form for each element.	[STIFF]
Assemble tri-block, tri-diagonal system:	[TRIASSEM]
Assemble $\mathbf{M}$ and $\mathbf{K}$.	
Form $\mathbf{C}$, then $\mathbf{K}^*$.	
Comute $\mathbf{R}^V$, $\mathbf{R}^M$.	
If first substep in step, compute $\mathbf{R}^D$.	

(4) Convergence? Load Applied?

Compute equilibrium, looping over all elements:	[EQBM]
Calculate $\mathbf{R}^U$, hence $\mathbf{R}_I^U$ and η.	
Set $\Delta \mathbf{R} = \mathbf{R}^U$.	

Print or save substep output. [OUTPUT]
If (η < tolerance) and ($f_{apl} \geq 1$), **exit substep loop**.

(5) Advance solution.
Solve for $\Delta\mathbf{r}$ in $\mathbf{K}^* \Delta\mathbf{r} = \Delta\mathbf{R}$. [SOLVE]
Compute f_{min} as governing minimum event factor [EFACT]
Advance solution by event factor: [FACTOR]
Scale $\Delta\mathbf{r}$ by f_{min}.
Calculate $\Delta\dot{\mathbf{r}}$ and $\Delta\ddot{\mathbf{r}}$.
Update $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$.
Update f_{rem}, f_{apl}

End substep loop, allowing for non-converged solution.

4.2.1.2 Algorithm for A Condensation to Super-Element. At the end of each converged local solution step, we may desire to condense the cable submodel into a super-element. The basic algorithm for doing this is as follows:

Compute super-element properties: [SUPERELM]
Apply $\mathbf{R}^S$ as a load, calculate $\mathbf{r}$ using $\mathbf{K}^*$ and hence get $\mathbf{Q}^S$.
Apply $\mathbf{R}^V$ as a load, calculate $\mathbf{r}$ using $\mathbf{K}^*$ and hence get $\mathbf{Q}^V$.
Apply $\mathbf{R}^M$ as a load, calculate $\mathbf{r}$ using $\mathbf{K}^*$ and hence get $\mathbf{Q}^M$.
Compute $\mathbf{K}^Q$ by condensing $\mathbf{K}^*$.
Compute $\mathbf{R}^Q$ by condensing $\mathbf{R}^E$.

4.2.2 Special Solution of Linear Matrix Subsystem. The element connectivity pattern for a cable submodel is unique and simple. Each element connects sequentially to the next element. The system matrices that result have all terms neatly grouped very close to the diagonal.

The number of finite cable elements in the cable submodel is one less than the number of nodes. The number of degrees of freedom at each node is three. Each node connects only one or two cable elements plus nodal elements. This simple connectivity produces a Jacobian matrix that is positive-definite, symmetrical, and tri-banded.

Tri-banded means that three sets of submatrices lay along the diagonal of the fully assembled Jacobian matrix. Each (3 by 3) submatrix corresponds to the three degrees of freedom at each node.

$$\begin{bmatrix} \hat{\mathbf{K}}_1 & \check{\mathbf{K}}_1 & & & & & \\ \check{\mathbf{K}}_1 & \hat{\mathbf{K}}_2 & \check{\mathbf{K}}_2 & & & & \\ & \check{\mathbf{K}}_2 & \hat{\mathbf{K}}_3 & . & & & \\ & & . & . & . & & \\ & & & . & \hat{\mathbf{K}}_{N-2} & \check{\mathbf{K}}_{N-2} & \\ & & & & \check{\mathbf{K}}_{N-2} & \hat{\mathbf{K}}_{N-1} & \check{\mathbf{K}}_{N-1} \\ & & & & & \check{\mathbf{K}}_{N-1} & \hat{\mathbf{K}}_N \end{bmatrix} \begin{bmatrix} \Delta\mathbf{r}_1 \\ \Delta\mathbf{r}_2 \\ \Delta\mathbf{r}_3 \\ . \\ \Delta\mathbf{r}_{N-2} \\ \Delta\mathbf{r}_{N-1} \\ \Delta\mathbf{r}_N \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{R}_1 \\ \Delta\mathbf{R}_2 \\ \Delta\mathbf{R}_3 \\ . \\ \Delta\mathbf{R}_{N-2} \\ \Delta\mathbf{R}_{N-1} \\ \Delta\mathbf{R}_N \end{bmatrix} \tag{4.15}$$

Where:

N = number of nodes

$$\mathbf{K}_i = \begin{bmatrix} K_{XX} & K_{XY} & K_{XZ} \\ K_{YX} & K_{YY} & K_{YZ} \\ K_{ZX} & K_{ZY} & K_{ZZ} \end{bmatrix}_i \qquad \Delta\mathbf{r}_i = \begin{bmatrix} \Delta r_X \\ \Delta r_Y \\ \Delta r_Z \end{bmatrix}_i \qquad \Delta\mathbf{R}_i = \begin{bmatrix} \Delta R_X \\ \Delta R_Y \\ \Delta R_Z \end{bmatrix}_i$$

$i = 1,2,3,\ldots,N$

The tri-bands contain all the non-zero terms in the full Jacobian matrix. Instead of storing the large sparse matrices of equation, we store only the smaller, denser submatrices. We store each set of submatrices (e.g., stiffness, damping, or inertia), as arrays with dimension equal to (3 by 3 by (# of nodes)).

We use a special algorithm for solving the linear matrix subsystem of algebraic equations (4.15) at each substep. This special algorithm works directly with the tri-banded data storage scheme. The algorithm involves the normal Gaussian elimination loops for decomposition, forward substitution, and back substitution but organized in a tri-banded block or hyper-matrix form.

$$\Delta\mathbf{r}_1 = \left(\hat{\mathbf{K}}_1\right)^{-1} \Delta\mathbf{R}_1$$

DO $i = 2, N, 1$ ← loop for decomposition & forward subsitution

$$\begin{aligned} \tilde{\mathbf{K}}_i &= \mathbf{K}' \check{\mathbf{K}}_{i-1} \\ \mathbf{K}' &= \hat{\mathbf{K}}_i - \check{\mathbf{K}}_{i-1} \tilde{\mathbf{K}}_i \\ \Delta\mathbf{r}_i &= \left(\mathbf{K}'\right)^{-1} \left(\Delta\mathbf{R}_i - \check{\mathbf{K}}_{i-1} \Delta\mathbf{r}_{i-1}\right) \end{aligned} \tag{4.16a}$$

↵

DO $i = (N-1), 1, -1$ ← loop for back subsitution

$$\Delta\mathbf{r}_i = \Delta\mathbf{r}_i - \tilde{\mathbf{K}}_{i+1} \Delta\mathbf{r}_{i+1}$$

↵

The inverse of the pivotal (3 by 3) submatrix is a simple algebraic expression (Tuma, 1970). The tri-band algorithm uses about as many floating point operations as a "skyline" algorithm. Both algorithms use significantly fewer operations than traditional Gaussian elimination.

$$\#\text{ of operations} \approx \begin{Bmatrix} \dfrac{(6N)^3}{3} + \dfrac{(6N)^2}{2} \\ 70(N-1) \end{Bmatrix} \text{for} \begin{Bmatrix} \text{full Gaussian} \\ \text{tri - banded Gaussian} \end{Bmatrix} \qquad (4.16b)$$

Where:

N = number of nodes (with three degrees of freedom per node)

Unlike the full Gaussian algorithm, the tri-band algorithm has no pivoting. Even if the matrix is nonsingular, the tri-band algorithm can theoretically fail because of numerical imprecision. If necessary, we can improve stability by using a procedure in which we decompose the 3 by 3 submatrices into lower and upper triangular submatrices at each sequential node and allow for submatrix permutations. However, we have not detected any significant numerical imprecision that would warrant a more stable procedure.

4.2.3 Boundary Conditions. The cable submodel can easily accommodate any type of boundary condition including the following types:

- Rigid fixity (infinite stiffness) with imposed displacement
- Variable fixity (finite stiffness) with imposed displacement
- Fluid surface interaction with associated variable buoyancy resistance
- Seafloor interaction with stiffness resistance and slope changes
- Other surface interaction such as cable-to-cable contact

For rigid fixity, we simply set displacement of the boundary nodes equal to the boundary displacement. We then eliminate the corresponding degrees of freedom from the matrix system solution.

For variable fixity, we establish a set of restraint springs attached between the boundary nodes and ground. We impose displacements at the boundary nodes by setting the reference location for the restraint springs to an offset value that equals the imposed displacement. The corresponding resisting force has a magnitude proportional to the imposed displacement. The resulting force imbalance automatically goes away when the node moves the required distance.

Carefully contrived, restraining springs can represent all types of surface interaction. Fluid surface interaction involves a vertical restraint spring that has a resisting force proportional to displaced fluid volume. Seafloor interaction involves a restraint spring normal to the seafloor that has resisting force proportional to displacement into the sea-bottom. Changes in seafloor slope affect the vector direction of the resisting force.

Event control is effective for maintaining solution stability in the presence of these discretely changing boundary conditions. This includes initial contact with the boundary,

changes in boundary elasticity, and separation from the boundary. In addition, events are important for identifying discrete changes in the slope of the seafloor.

4.2.4 Redefinition of Submodel. Traditional nonlinear solution schemes put the state determination module at the end of the step. Consequently, to begin the step loop we must have a pre-defined state. This order makes it difficult to redefine the submodel.

Our chosen nonlinear solution strategy determines state at the beginning of each step, immediately after step initialization. This re-ordering of the computational modules allows for easy progressive redefinition of the submodel. By simply changing a few variable properties of the element in the step initialization module, we can easily accommodate any redefinition of the submodel. The normal state determination module, which immediately follows the step initialization module, automatically computes all required state variables for the redefined submodel configuration.

Cable paying involves the following types of submodel redefinition at every step:

- Evolving boundary conditions
- Different makeup of nodes and elements
- Changes in element connectivity
- Changes in element stiffness, damping, and mass

Cable pay-out means adding rope to either end of the cable. Similarly, cable pay-in means removing rope at either end of the cable.

With the chosen cable submodel and chosen solution strategy, the cable paying task simply involves increasing or decreasing the slack/taut length of the end element connected to the paying node. There is no need for cumbersome readjustment of all state variables and system matrices to facilitate cable paying, as required in the traditional SEADYN algorithms (Webster, 1982).

To maintain a reasonable balance of element lengths all along the cable submodel, it is advisable to add or remove elements from the submodel when element lengths become too large or too small. When the slack/taut length of the end element reaches twice the nominal slack/taut element length, we add an element using the following simple tasks:

- Divide the slack/taut length of the long element into two equal lengths.
- Increment the counter that tracks the number of nodes and elements.
- Assign coordinate position, velocity and acceleration to the new node.

The dynamic response of the cable at the paying node is very sensitive to the algorithms used for cable paying (Leonard and Karnoski, 1990). We choose to use a simple average of the coordinates, velocities, and accelerations of adjacent nodes and assign these median values to the new node as shown in Figure 4.4. Our choice produces expected dynamic results with remarkable stability.

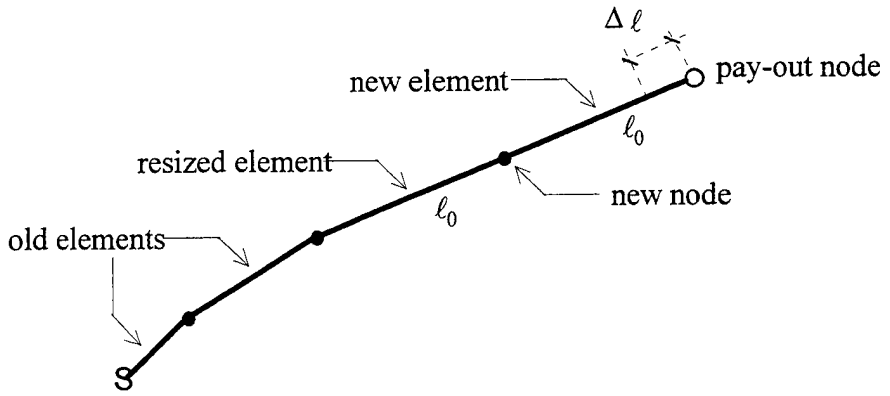


Figure 4.4 Cable pay-out creating a new element.

Where:

ℓ_0 = nominal slack / taut element length

$\Delta\ell$ = incremental length of cable payed out during step

When the slack/taut length of the end element reaches one-half of a nominal slack/taut element length, we remove an element using the following simple tasks:

- Combine the slack/taut length of the last two elements.
- Decrement the counter that tracks the number of nodes and elements.
- Zero-out coordinate position, velocity, and acceleration for the removed node.

We assign zero values to all inactive element properties, state variables, and nodal variables. Only active nodes and elements participate in the nonlinear solution.

4.3 Special Solution Controls

When based on real structural behavior, event control helps maintain a level of numerical stability for the model that approaches the physical stability of the structure itself. A nonlinear solution strategy with event control eliminates the traditional need to "ramp" the load at the beginning of the simulation. In other words, there is no need for gradual application of the load to reach a stable and meaningful solution. The simulation results are stable and real from the very beginning of the simulation.

Dynamic steps can follow static steps and vice versa. No special order of static and dynamic steps is necessary. Consequently, the system need not be in static equilibrium to perform a dynamic step. We also need no special procedure to "bring the system to rest." We simply execute a static step.

A good check for numerical stability is to establish a steady-state response (with no damping terms) and see if this response persists for all time. If there are modeling errors, numerical noise will develop and eventually cause the solution to diverge.

4.3.1 Axial Event Factor Calculation. There are two possible axial events for each finite cable element for which we must calculate an event factor.

- The current element state is taut, and the element may go slack.
- The current element state is slack, and the element may go taut.

Calculation of the axial event factor requires a special procedure for precise event control. With large displacements and rotations of an element, an event determined by an axial limit within the element does not linearly relate to displacements at the element nodes. Figure 4.5 shows the nonlinear relation between element axial deformation and nodal displacement.

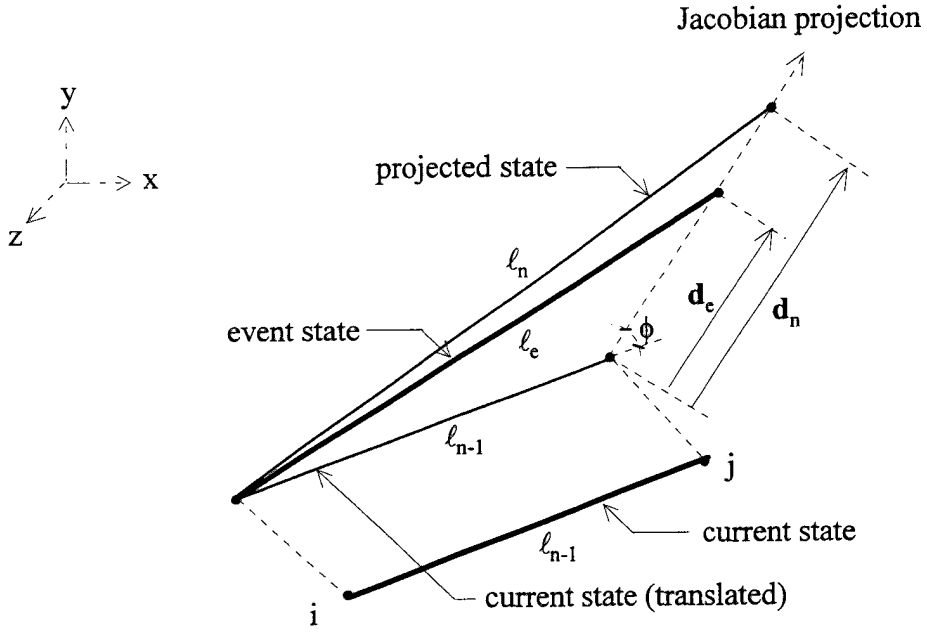


Figure 4.5 Axial event for increasing element length.

Where:

i, j = element nodes

x,y,z = coordinates

$\ell_{n-1}, \ell_e, \ell_n$ = element length for current, event, projected states

$\mathbf{d}_n$ = incremental displacement for projected state

$\mathbf{d}_e$ = incremental displacement for event state

ϕ = angle between current element axis and Jacobian projection

The incremental displacement for the projected state results directly from the solution of the step-linear (Jacobian) system. If in moving to this projected state, the element triggers an axial event, we seek a smaller increment of displacement, one scaled back to where the axial event occurred. To scale back the projected solution, we compute an event factor defined as the following ratio of vector lengths.

$$f = \frac{D_e}{D_n} \quad \text{with the following restriction:} \quad 0 \le f \le 1 \tag{4.17}$$

Where:

$D_e = |\mathbf{d}_e|$ = length of displacement vector for event state

$D_n = |\mathbf{d}_n|$ = length of displacement vector for projected state

The incremental displacements relate directly to the total displacement.

$$\begin{aligned} \mathbf{d}_n &= \left(\mathbf{r}_{j_n} - \mathbf{r}_{j_{n-1}}\right) - \left(\mathbf{r}_{i_n} - \mathbf{r}_{i_{n-1}}\right) \\ \mathbf{d}_e &= \left(\mathbf{r}_{j_e} - \mathbf{r}_{j_{n-1}}\right) - \left(\mathbf{r}_{i_e} - \mathbf{r}_{i_{n-1}}\right) \end{aligned} \tag{4.18}$$

The Pythagorean theorem gives the following simple relationship of lengths for the event and current states of the element.

$$\ell_e^2 = \left(\ell_{n-1} + D_e \cos\phi\right)^2 + \left(D_e\left(1 - \cos^2\phi\right)\right)^2 \tag{4.19}$$

Where:

$$\cos\phi = \frac{1}{D_n}\mathbf{d}_n^T\mathbf{c}$$

Quadratic equation (4.19) has two roots representing element lengths for two different event-controlled displacement vectors along the Jacobian projection. The minimum positive root represents the desired displacement vector.

$$D_e = min\left\{-b \pm \sqrt{\frac{b^2 - 4ac}{2a}}\right\} \geq 0 \qquad (4.20)$$

Where:

$$a = cos^4\phi - cos^2\phi + 1$$
$$b = 2\ell_{n-1}cos\phi$$
$$c = \ell_{n-1}^2 - \ell_e^2$$

In Figure 4.5, the Jacobian projection increases the element length. In this case, the negative root of equation (4.20) represents a displacement vector that is in the wrong direction. We thus choose the positive root. Alternatively in Figure 4.6, the Jacobian projection reduces the element length. In this case, the larger positive root of equation (4.20) represents a displacement vector that occurs beyond the first desired displacement vector. We thus choose the smaller positive root.

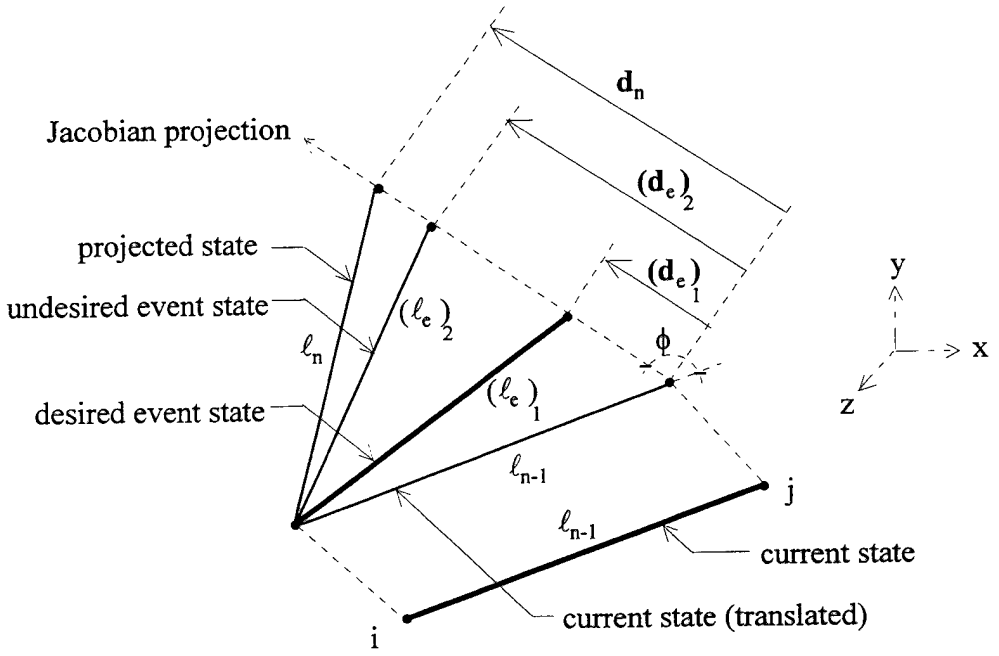


Figure 4.6 Axial event for decreasing element length.

The following restriction applies to the radical of equation (4.20):

$$0 \leq b^2 - 4ac \qquad (4.21)$$

This restriction represents a Jacobian projection where an axial event will not occur. In other words, no matter how large the displacement increment is, no axial event for this element controls the solution in the current substep.

Instead of stopping the solution exactly at an event, the solution should overshoot the event by a very small force tolerance. By overshooting the event, the solution avoids re-triggering that same event. This also assures that an abundance of closely spaced events do not adversely hold up the overall speed of simulation. Overshooting the event by a large amount can cause loss of numerical stability, because this allows the solution to deviate from the true solution path.

At the element level, the overshoot factor should have a specific physical meaning. For the cable finite element, we choose the overshoot tolerance as a very small percentage of the given slack/taut tension.

4.3.2 Energy Balance. Final step convergence requires not only that the force unbalance be less than the force convergence tolerance, but also that the solution conserves energy. In nonlinear structural simulations, the system accumulates, dissipates, and transfers energy into various forms at each time step. In a nonlinear structure, the following physical effects can alter the energy balance:

- Loads acting through respective displacements
- Damping and inertial forces acting through respective motions
- Structure moving through a force field such as gravity
- Hysteretic and viscous damping dissipation
- Elastic strain potential

Energy balance computations are valuable for assessing solution accuracy. In a nonlinear system, it is possible to have more than one solution that will satisfy a desired force balance. A substantial imbalance of energy over time indicates a high probability that the particular solution chosen is incorrect.

Energy balance depends on the basis we choose for computing energy. For example, the potential energy associated with gravity depends on our choice of elevation reference. We choose the initially defined position of each node as the reference elevation and thus zero potential energy with respect to gravity.

For the cable finite element, we choose the slack/taut state of each cable finite element as the best base for zero elastic energy. Given a very small slack/taut tension, the elastic energy in each element is basically the product of its axial deformation and action.

We also choose to start all simulations from rest. In other words, we set all velocities and accelerations initially equal to zero. This provides an excellent base from which to compute all dynamic energy changes. This also eliminates starting conditions that may be inconsistent with the physical nature of the structure and avoids chaotic sensitivity to initial starting conditions.

Initial conditions are necessary for starting a numerical model. A real structure has no simple initial conditions but rather a complex evolving state that we only begin to quantify with initial conditions for a simple nonlinear numerical model. No matter how complex a nonlinear model we use, it is still rather simple in comparison to the complexity of the real structure that we are trying to simulate.

Energy is a popular norm for determining how to improve the nonlinear solution. However, a simple energy norm ignores the complex spatial nature of energy generation. It also ignores the spatial gradient of all of these energy sources. As such, it is not a good method for directing the nonlinear solution. Rather, it serves simply as a way to check for complications in the solution that may indicate that the computed solution is not the desired solution. Event control is the preferred strategy for stable nonlinear solution of compliant structural systems.

5. NEW COMPUTER CODE

For testing the new modeling theories and the nonlinear solution, we choose a two-pronged effort. We implement new elements for an existing general-purpose finite-element code (DRAIN). We also generate a new computer code (MBDSIM) that is specific for simulation of compliant marine structures.

5.1 Object-Oriented Implementation

At the heart of most traditional finite element codes is a single-domain computational framework. It is the "glue" that holds the computer code together. Consequently, it is difficult to modify most existing computer codes for local/global computations, unless the code is object-oriented.

The DRAIN family of codes (Prakash and Powell, 1993) has a modular element interface that is rather object-oriented. Modular element objects perform specific element actions on specific element data. A set of well-defined FORTRAN subroutines performs the actions. A well-structured FORTRAN common block stores the data. Each modular element has a standard action and data interface connecting the element object to the DRAIN base program. The base program is responsible for managing the data files, processing data associated with the "global" model, and obtaining the "global" solution.

With the current object-oriented interface, we can easily add new elements and even simple static super-elements to the DRAIN base program. With future modification of this interface, we will be able to add complex dynamic super-elements to the DRAIN base program. With this new object-oriented interface, regular elements and super-elements (collectively called element objects) would have identical action and data interfaces. As a super-element, each element object would privately generate its local submodel, implement its local solution, and convert itself to a super-element.

5.1.1 Action Interface. Given an element object, a set of well-defined FORTRAN subroutines performs the object actions. For the new super-element, these actions are similar to those for the traditional finite element. The following list summarizes the actions for the current element object (Powell, 1995):

- Input and initialize element data.
- Set element initial states.
- Form equivalent end forces for element loads (if any).
- Form equivalent end forces for element inertia/damping (if any).

- Calculate element stiffness.
- Calculate element event factor.
- Update element state.
- Save and print element time histories.

BASE PROGRAM			ELEMENT SUBROUTINES	
PRIVATE	*INTERFACE*			*PRIVATE*
Memory allocation		<<<	Memory required	
# of element groups	Group data (flat block)	<=>	Group data (structured)	Group data-structure
Data files	File control	>>>		Group properties
# of elements	Element data (flat block)	<=>	Element data (structured)	Element data-structure
# of DOF	Element assembly	<<<	Elem. connectivity	Element properties
Mass matrix		<<<	Element event	Details of event
Stiffness matrix		<<<	Element stiffness	
Nodal load		<<<	Element energy	Element state
Nodal mass		<<<	Elem. event factor	Element theory
Inertia force		<<<	Elem. resist. force	
Viscous damping force		<<<	Initial stiffness for linear damping	
Nodal restraints	Nodal coordinates	>>>	Element position	
Solution strategy	Nodal velocity & displacement	>>>	Element movement	Element deformation

Figure 5.1 Data interface for typical element object.

5.1.2 Data Interface. Figure 5.1 depicts data privately held by the element object and data publicly exchanged across the interface between the base program and the subroutines of an element object. Both traditional finite elements and new super-elements must meet the requirements of this data interface.

Figure 5.2 depicts the additional items necessary for extending the data interface of Figure 5.1 to include super-element objects. Most of the new data items represent data for computing local loads that result from velocity and acceleration fields defined at the global level. These loads, such as gravity, buoyancy, and fluid drag, can be a variety of types and can have a variety of effects on the resulting super-element. All super-elements must meet the additional requirements given in this extended data interface.

BASE PROGRAM			**ELEMENT SUBROUTINES**	
PRIVATE	*INTERFACE*			*PRIVATE*
		<<<	# of elm. load sets	
		<<<	Field load flag	
		<<<	Field location	
Details of accel. & velocity fields	Field values at given locations	>>>		Details of submodel load sets
Factors for submodel load sets	Load scale factors	>>>		Theory for field & submodel load
		<<<	Force increment at global nodes	Details of local submodel
	Time & time step	>>>		
		<<<	Condensed Jacobian	Response of local nodes
		<<<	Condensed unbalanced load	Local submodel solution strategy
	Nodal velocities & accelerations	>>>		
	Submodel control parameters	>>>		Meaning of control parameters

Figure 5.2 Extension to data interface to accommodate super-elements.

5.2 Elements for Existing Base Program

Since DRAIN-2DX and DRAIN-3DX had no elements for properly representing cables, we create two new element objects, a finite cable element and a semi-analytical cable super-element. Both elements have a rudimentary representation of nonlinear dynamics consistent with limitations on dynamic analysis in DRAIN. These limitations include constant nodal mass, consistent nodal loads, and diagonal damping (based on initial system stiffness).

Consistent with the event-by-event solution strategy in DRAIN, element events tell the base program when to reformulate stiffness. Since DRAIN has no direct provision for iteration on force unbalance, these discrete element events take on a great importance for both stability and accuracy of the solution.

5.2.1 Finite Cable Element. We implement the finite cable line element based on modeling principles described in Section 3.3. By changing a single variable in the top level common block, the element becomes either a two- or three- dimensional element for respective use in either DRAIN-2DX or DRAIN-3DX. The cable line element can take any orientation in space. Each element has a node at each end of the element. Each node has translational degrees of freedom in each coordinate direction as shown in Figure 5.3.

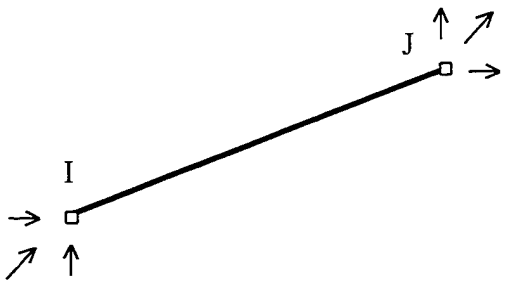


Figure 5.3. Finite cable element.

This element has three types of events as given in the table below:

Code	Event type
-1	Element goes taut
0	No event
+1	Element goes slack
+2	Rotation of element axis exceeds specified tolerance

If a non-zero event occurs, DRAIN restrains the solution to an element deformation that just overshoots this event. Element deformation is the distance between nodes. For simplicity, the magnitude of deformation for overshooting the event is a small percentage of the slack/taut element length.

5.2.2 Semi-Analytical Cable Super-Element. We implement the semi-analytical cable super-element based on modeling principles described in Section 3.2. This super-element has two end nodes and represents the two-dimensional planar behavior of a simple cable catenary between these nodes. The element's catenary behavior is dependent on a non-zero distributed load (e.g., self-weight) acting along the entire length of the cable. The element will automatically orient itself in the plane that contains the two end nodes and the distributed load as shown in Figure 5.4. The magnitude and direction of this distributed load can change with time.

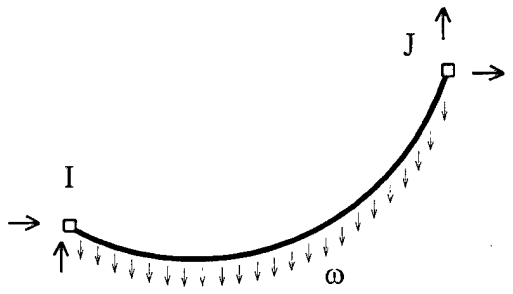


Figure 5.4. Semi-analytical cable super-element.

Since the cable catenary element exists only in a two-dimensional plane, we develop the element for use in DRAIN-2DX only. For use in DRAIN-3DX, we must add representation of element response in the third dimension out of the catenary plane.

This super-element possesses the same element functionality and interacts with the base program of DRAIN-2DX in much the same way as any traditional element. To determine element state (shape, end forces, and stiffness), this super-element has a specialized local iterative strategy for solving a set of semi-analytical nonlinear catenary expressions as described in Section 3.2.

The stiffness of the cable catenary changes continuously with deformation. A cable catenary exhibits small stiffness when slack and high stiffness when taut. This super-element has three types of events as given in the table below:

Code	Event type
-1	Cable goes taut
0	No event
+1	Cable goes slack
+2	Zero horiz. tension

Zero horizontal tension occurs when node "I" and "J" are vertically above one another. If one of the non-zero events occurs, DRAIN restrains the solution to an element deformation that just overshoots the event. Element deformation is the distance between nodes. For simplicity, the magnitude of deformation for overshooting the event is a small percentage of unstressed cable length along the catenary.

5.3 Multi-Body Dynamic Simulation

MBDSIM (Multi-Body Dynamics Simulator, version 2.0) is a fairly general, independent computer code that can model many kinds of nonlinear structures. In particular, this includes compliant marine structures that consist of one cable substructure and its interactions with other nonlinear marine substructures, such as buoys, seafloor plows, and surface vessels. To represent the cable substructure, MBDSIM implements the numerical cable super-element (described in Section 3.4).

5.3.1 Overall Modeling Options. MBDSIM offers two major modeling options. MBDSIM can either act as a local submodel dependent on a base program (DRAIN) or as a global model independent of any base program.

5.3.1.1 Local Submodel Option. With this option, MBDSIM solves only the local cable submodel and generates a global super-element representation of the cable submodel. Using this super-element, an appropriate base program, such as DRAIN-3DX, can connect other super-elements and effectively simulate any kind of compliant marine structure.

The chosen cable submodel consists of several finite cable elements singly connected by local nodes as shown in Figure 5.5. The cable submodel may have several local nodes but only two global nodes, one at each end of the cable submodel. Any node may have a nodal element that interacts with a fixed or moving boundary.

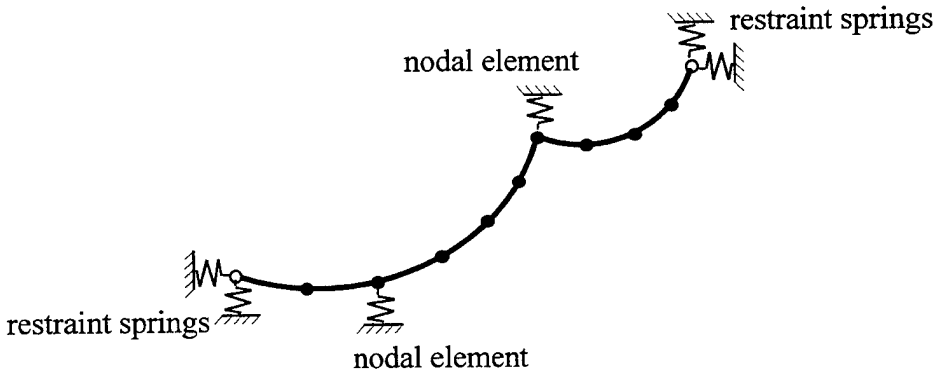


Figure 5.5 Local sub-model option.

At each step of the simulation, the base program may specify global increments of displacement at the two global nodes. MBDSIM imposes the specified displacements by either a

direct or an indirect method. The direct method involves specifying the displacement of the end node and then eliminates the corresponding degrees of freedom from the matrix system solution.

The indirect method involves a set of "very stiff" restraint springs at each end node. We impose displacements at the global nodes of the cable super-element by setting the reference location for the restraint springs to an offset value that equals the imposed displacement. This automatically creates a force unbalance that goes away when the node moves the required distance.

5.3.1.2 Global Model Option. With this option, MBDSIM solves a global model consisting of a single cable submodel with optional submodels attached to each end of the cable submodel. MBDSIM allows a seafloor plow submodel and a tow-ship submodel at the bottom and top ends of the cable, respectively, as shown in Figure 5.6.

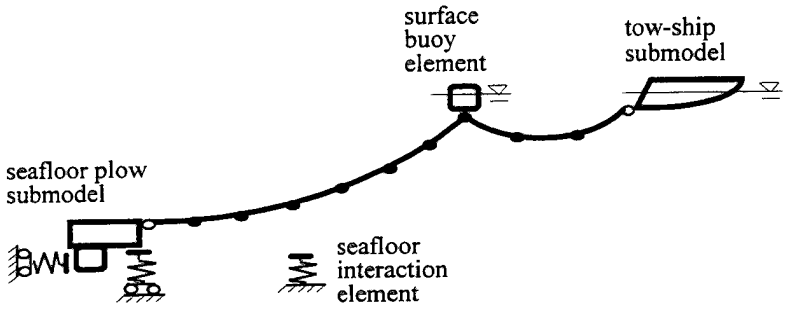


Figure 5.6 Global model option.

Currently, the seafloor plow submodel is a simple seafloor plow element as described in Section 3.6. The details of the tow-ship submodel are proprietary. Each node of the cable submodel may have a nodal element that connects the node to a specified boundary. These nodal elements can represent hydrostatic buoyancy or boundary contact at any given node of the cable. Figure 5.6 shows two nodal elements, a simple seafloor interaction element and a simple surface buoy element (described in Section 3.5).

The data storage scheme in MBDSIM requires that each nodal element connect strictly to only one node with only three translation degrees of freedom. If the boundary condition requires more degrees of freedom, a full boundary submodel is advised.

5.3.2 Computational Performance. Finite cable elements and nodal elements make up an excellent submodel for most cable substructures. Events in the new cable finite element are effective in dealing with slack/taut conditions within the cable. Similarly, events in the nodal elements are effective in dealing with changing boundary conditions.

For a realistic starting condition, we choose starting motion of all nodes to be "at-rest," namely zero velocity and zero acceleration. For simple generation of local nodes, we choose a starting configuration for the cable submodel, whereby all local nodes are equidistant along a

straight line between the end nodes of the cable submodel. We also choose to set initial tensions in each cable finite element to an equal slack or taut value.

The new nonlinear solution strategy with event control is very effective in controlling force unbalances. As such, the numerical solution practically follows the exact response path. This path is the correct resistance for every applied load at every step. Consequently, the step-by-step simulations generated by MBDSIM are extremely robust, i.e., fast, efficient, and stable.

Accelerations and velocities can vary quite dramatically during substep iterations due to sudden force imbalances resulting from rapid changes in stiffness. These numerically induced changes in velocities can cause unrealistic spikes in viscous forces at any given substep. This is particularly true for hydrodynamic forces that depend on the square of velocity.

Since we do not have force equilibrium until the end of a step, motions at each substep are unrealistic. Therefore, we choose to base viscous forces only on velocities at the end of the steps. Holding viscous forces constant during a step greatly improves solution efficiency. Since a typical time step is small in comparison to the time it takes for real viscous forces to change over an element, this assumption is acceptable.

5.3.3 Modeling Instabilities. Only a few clearly identifiable modeling instabilities remain within MBDSIM. For stable solution, all forces that depend on structure motion should be part of the Jacobian projection. This projection is the left-hand side of the equations of motion. The present engineering representations for hydrodynamic and frictional forces do not allow these computations to fit naturally in the linearized Jacobian projection. Instead, we must add increments of these forces to the unbalanced force. This force is the right-hand side of the equations of motion. Doing this compromises nonlinear solution stability.

5.3.3.1 Poor Hydrodynamic Representation. MBDSIM computes hydrodynamic drag forces using a relative motion formulation of the traditional Morison equation (described in Section 2.2.2.3). The original purpose of this equation was to compute the maximum hydrodynamic force that a solitary ocean wave imposes on a differential piece of a fixed, vertical, and rigid cylinder. Analysts now routinely use the equation to compute time-varying hydrodynamic force on an assembly of moving, inclined, flexible, non-cylindrical elements. This purpose is well beyond the intended limits of the original equation and its empirical coefficients.

Morison equation also computes force components that are rotationally inconsistent. For example, given a horizontal current velocity acting on a vertical finite cable element, the Morison equation gives the correct horizontal force. However, given the same horizontal current velocity acting on an inclined finite cable element, the Morison equation gives a vertical force in addition to the expected horizontal force.

Because of this rotational inconsistency, we need too many substep iterations to resolve the hydrodynamic force imbalance. This rotational inconsistency is more apparent in MBDSIM than traditional models, because MBDSIM (with its superior robustness) can compute much larger increments of rotation in any substep than traditional cable models.

Researchers propose other engineering formulations of hydrodynamic loading that attempt to resolve problems with the Morison equation (Chakrabarti, 1990). However, empirical coefficients for these new equations are generally not available. MBDSIM needs a better engineering equation for computing hydrodynamic force. We should design this new equation

specifically for computing step-linear hydrodynamic forces on a moving, rotating, and flexible finite element.

5.3.3.2 Poor Frictional Representation. MBDSIM computes frictional forces for surfaces in contact with a boundary using the Coulomb friction equation (described in Section 3.6.1). The original purpose of this equation was to compute only the maximum static resisting force between two ideally rigid contact surfaces. Analysts now routinely use this equation to compute time-varying, dynamic frictional forces on less-than-ideal, flexible contact surfaces. This purpose is well beyond the intended limits of the original equation and its empirical coefficient. The Coulomb friction equation is a gross over-simplification of true frictional behavior and thus compromises nonlinear solution stability.

We need a better engineering equation for computing frictional forces. MBDSIM needs a new type of frictional element that determines time-varying frictional resistance as a unique function of deformation (or displacement), not as a function of normal force. This new type of frictional element would be more consistent with our chosen displacement-based theory of structural analysis than is the traditional force-based Coulomb equation.

6. TEST PROBLEMS

Solution instability is a natural consequence of having to use a step-linear solution strategy to seek solutions for a nonlinear structural system. It is impossible to prove numerical stability for any new nonlinear solution strategies. Consequently, we resign ourselves to numerically demonstrating adequate efficiency, speed, and stability using a set of test problems.

Many classical test problems for compliant structures are available including a comprehensive and practical series of large-scale cable experiments (Palo, 1983). Rather than reproducing these classical results, we select test problems that traditionally have had poor or even non-existent numerical solutions.

Being bold, we pose test problems near the edge of stability. If the solution strategy works for these problems, it certainly will work for typical problems well within the domain of stability. We refer to this method of testing nonlinear solution models as: "walking along the edge of the canyon of instability without falling into the canyon."

In each test problem, we try to demonstrate a specific example of significant improvement in nonlinear modeling capability for compliant structures. Some of these test problems have well-known results recently published in the technical literature, while the majority are of our own making.

Unless otherwise noted, we obtain all results using the MBDSIM code with event control. Unless otherwise noted, we choose the following nominal values for nonlinear solution control.

- Angle tolerance for the rotational event = 10 degrees
- Slack/taut tension for the deformation event = element weight
- Force tolerance for overshoot either event = one force unit
- Force tolerance for solution convergence = one force unit

One force unit equals one pound or one Newton, depending on whether we use English or metric units, respectively.

6.1 Nonlinear Pendulum

Large displacements and large rotations are a key part of the new finite cable element. An excellent test problem for studying the large rotations of the new finite cable element is a simple pendulum as shown in Figure 6.1. We can drop the pendulum from a horizontal position (Bathe, 1982) or fire it from a vertical position with an initial horizontal velocity (Crisfield, 1994). The resulting behavior is the same. Using our new finite cable element and its associated nonlinear solution strategy, we extend the traditional domain of stability for this test problem.

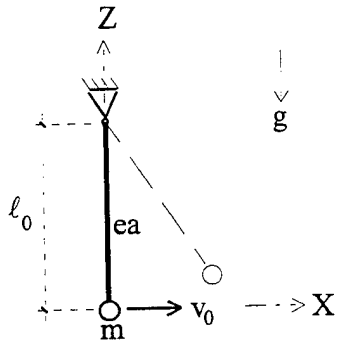


Figure 6.1 Test problem -- pendulum.

Where:

ea = 10^{10} N = stiffness factor

m = 10 kg = mass

ℓ_0 = 304.43 cm = unstretched cable length

g = 980 cm / s^2 = acceleration due to gravity

Δt = 0.015 s = time step

v_0 = initial velocity (given)

X,Z = coordinates

A traditional closed-form analytical solution exists for the test problem, if we linearize the problem using three simplifying assumptions:

- Use rotation as the principal unknown variable.
- Ignore stretch of the connecting rod.
- Restrict motion to very small angles.

Given these assumptions and restricting ourselves to very small initial velocities, the pendulum has the following static stretch and natural period of dynamic motion:

$$\Delta Z_{static} = -\frac{mg\ell_0}{ea} \approx -3x10^{-6}\,cm = \text{stretch due to gravity force}$$
$$T = 2\pi\sqrt{\frac{\ell_0}{g}} = 3.5\,s = \text{natural period (linearized pendulum)} \qquad (6.1)$$

With a finite element model of the pendulum, it is not necessary to make the previous simplifying assumptions. One finite element with axial elasticity is sufficient for characterizing the motions of the pendulum. Using one finite cable element, we set its natural slack/taut tension equal to the weight of the mass.

6.1.1 Swinging Motion. Given a small initial horizontal velocity of less than 772.5 cm/s, the mass swings about the central pivot. For an initial horizontal velocity equal to 772.5 cm/s, the mass swings to a full horizontal position. This nonlinear, semi-circular motion has a natural period of about four seconds. Using a traditional truss finite element (which supports axial compression), the motion describes a fairly sinusoidal function. Elastic stretch in the connecting rod flattens the sinusoidal peaks and valleys ever so slightly.

Conventional solution strategies require a time step of less than or equal to 0.025 seconds to achieve a satisfactory solution (Crisfield, 1994). An unsatisfactory solution is one where energy transfers rapidly from the desired pendulum motion to excessive axial vibrations. Using the new finite cable element, we obtain results consistent with the previously published results.

For an initial velocity equal to 772.5 cm/s, the finite cable element just goes slack. Consequently, we obtain a cyclic swinging motion for the mass that is not quite sinusoidal, but rather, has obvious flattened peaks and valleys, as shown in Figure 6.2. The degree of flatness depends on the magnitude of the very small tension of the cable finite element when it is slack. With no damping, the periodic swinging motion shown in Figure 6.2 continues unchanged for all time.

6.1.2 Whirling Motion. If we use larger initial velocities, we get much more interesting results than the previously published results. With an initial horizontal velocity exceeding 772.5 cm/s, the mass passes the horizontal position. Using a traditional truss finite element (which supports axial compression), the mass would continue to describe an arc about the central pivot, albeit a longer arc and even a complete circle. The transition from swinging motion to complete circular whirling motion is a physical bifurcation.

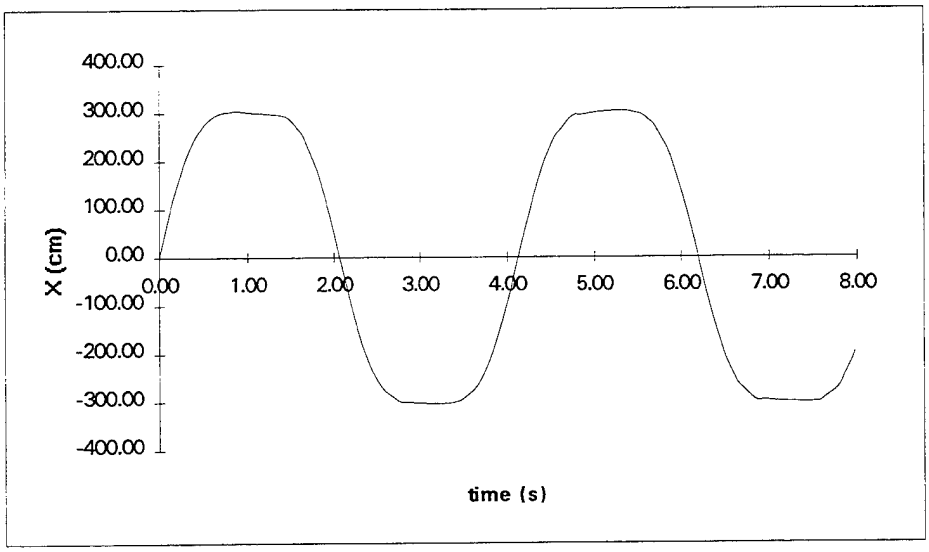


Figure 6.2 Swinging motion with slackness -- pendulum.

Using the new finite cable element (which can go slack), we need an initial horizontal velocity greater than about 3,000 cm/s to produce complete circular whirling motion. At decreasing initial velocities below this value, the cable element goes slack and the circular arc begins to flatten at its highest elevation.

Figure 6.3 shows what happens for an initial horizontal velocity equal to 2,250 cm/s. As the force of gravity acts to cancel the "centrifugal" force, the mass falls ever so slightly toward the central pivot. Afterwards, the mass continues to whirl but also bounces elastically against the confining limit of the taut cable. Since there is no damping in this test model, the whirling and bouncing motion will continue for all time. In reality, material damping may cause the bouncing motion to damp out after only a few bounces. Since we know little about the local physical mechanisms internal to a cable, we cannot yet accurately model the impulsive damping that results when a cable snaps taut.

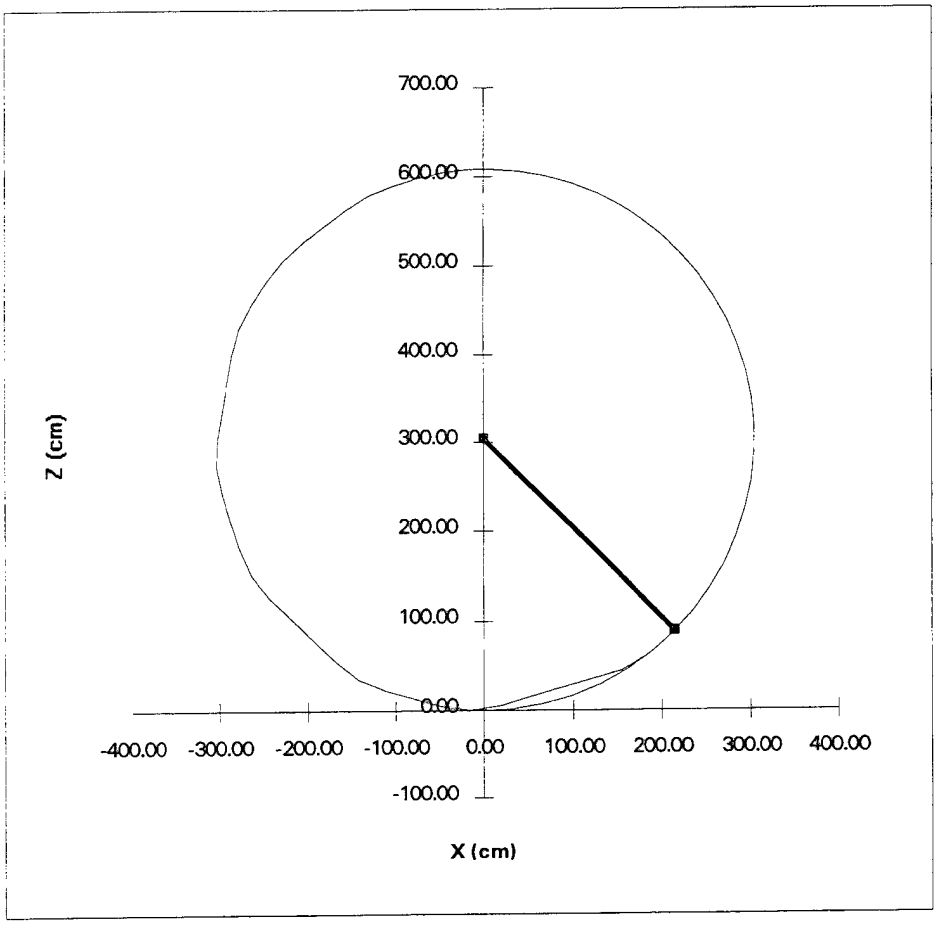


Figure 6.3 Whirling motion with bouncing -- pendulum.

6.1.3 Over-Swinging Motion. With an initial horizontal velocity equal to 850 cm/s, the mass behaves in a manner similar to a child swinging too high on a swing set. The cable goes slack and the child (i.e., mass) falls vertically down for a short distance, then recovers just to repeat the same snapping action on the other side.

Figure 6.4 shows this over-swinging motion. Figure 6.5 shows the sharp spikes in tension that result from the cable snapping taut.

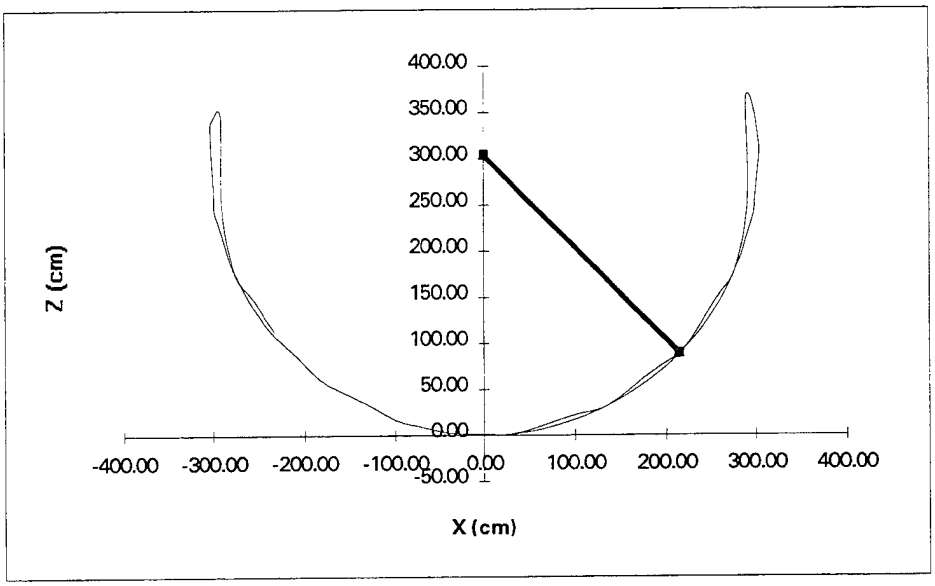


Figure 6.4 Over-swinging motion with snapping -- pendulum.

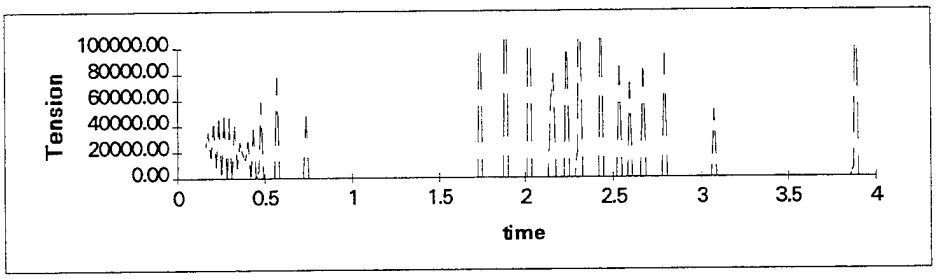


Figure 6.5 Over-swinging tension snaps -- pendulum.

6.2 Varying-Span Cable Catenary

A horizontally suspended cable with one support free to slide horizontally is an excellent test problem for studying the numerical robustness of a cable submodel. Given a horizontal load applied to the moving support, Figure 6.6 depicts the final equilibrium solution for this test problem. As shown, ten finite elements are sufficient for representing the problem. Using this highly nonlinear test problem, we show how robust our simple modified Newton-Raphson solution strategy is when compared to traditional nonlinear solution strategies.

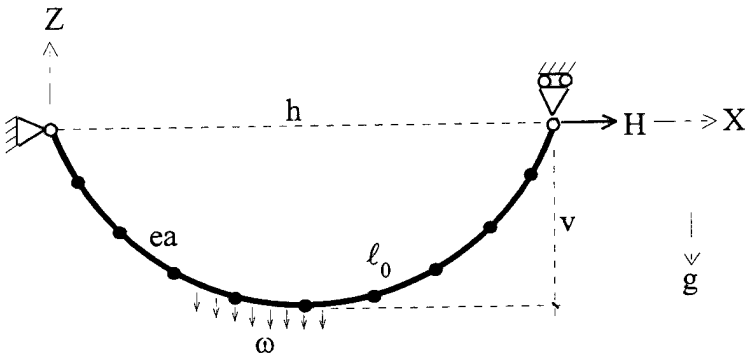


Figure 6.6 Test problem -- varying-span cable.

Where:

$ea = 10^5$ lb = stiffness factor

$\omega = 0.1$ lb / ft = weight per unit length

$\ell_0 = 200$ ft = unstretched length of entire cable

$H = 5.7735$ lb = horizontal load

$h = 152.2$ ft = final horizontal span

$v = 58.0$ ft = final vertical sag

$g = 32.2$ ft / s^2 = acceleration due to gravity

We solve the test problem for two test cases that encompass the two classical nonlinear solution difficulties:

- An ill-posed starting configuration
- An ill-conditioned solution path

6.2.1 Ill-Posed Starting Configurations. Traditional nonlinear solution strategies require a well-posed starting configuration, whereby the initial cable shape is consistent with the cable pretensions. One possible starting configuration that meets this traditional requirement is a straight line shape along the positive x-axis and pretension equal to the applied horizontal load. From this initial configuration, most traditional nonlinear solution methods converge to the correct answer in a reasonable number of iterations (Shugar, 1988). Using the new finite cable element and its associated nonlinear solution strategy, we obtain the correct solution with one full static load step containing as few as two iterative substeps.

For the same problem, a more challenging starting configuration is a straight line shape along the negative x-axis and no pretension. For this starting configuration, traditional Newton-based nonlinear solution strategies do not generally converge (Webster, 1980). A dynamic relaxation strategy does not converge even after 660 iterations, and a viscous relaxation strategy converges only after 28 iterations, at best (Webster, 1980). For the same problem, an automated

dynamic relaxation strategy guarantees convergence, but only after hundreds of iterations (Shugar, 1988).

Using the new finite cable element and its associated nonlinear solution strategy, we obtain the correct solution with one static load step containing only twelve iterative substeps. Ten of these substeps result directly from deformation events (slack/taut events) in each of the ten elements. Figure 6.7 shows the shape of the cable at each substep. With event control, the intermediate shapes have reasonable displacement, and thus force unbalance remains fairly small.

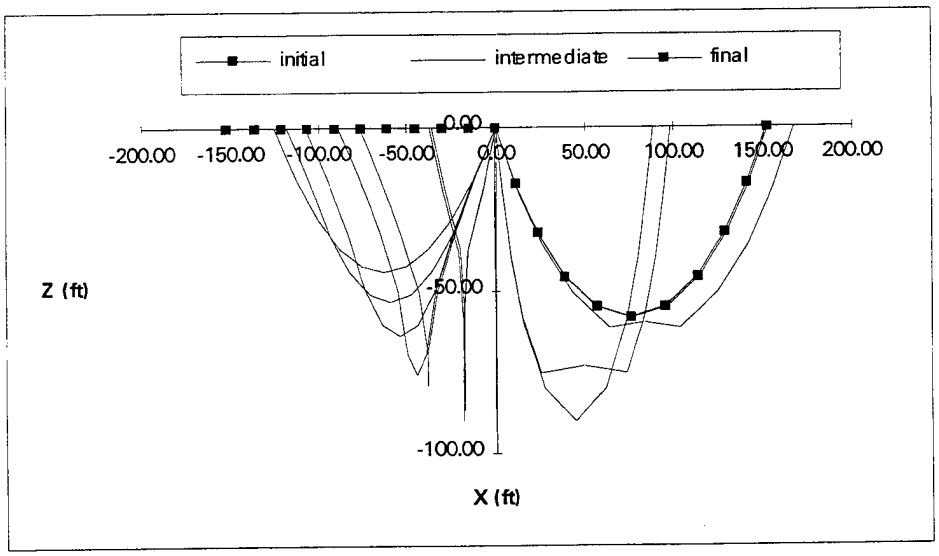


Figure 6.7 Substep solutions with event control -- varying-span cable.

Event control is not always necessary for stability in the nonlinear solution. The new finite cable element itself has all the stability needed for this particular problem. If we turn-off event control, we obtain the correct solution with one static load step containing only two iterative substeps. Figure 6.8 shows the starting (initial) shape, the only intermediate shape, and the final shape of the cable. Because we do not use event control in this solution, the intermediate shape has a very large displacement and thus a very large force imbalance. However, it only takes one iterative substep to correct this force imbalance and achieve the correct final solution.

For the test problem, we can pose many other starting configurations (Shugar, 1988). None of these starting configurations pose any difficulty for the new finite cable element and its associated nonlinear solution strategy. Any starting configuration, within obvious geometrical and tension realism, is permissible.

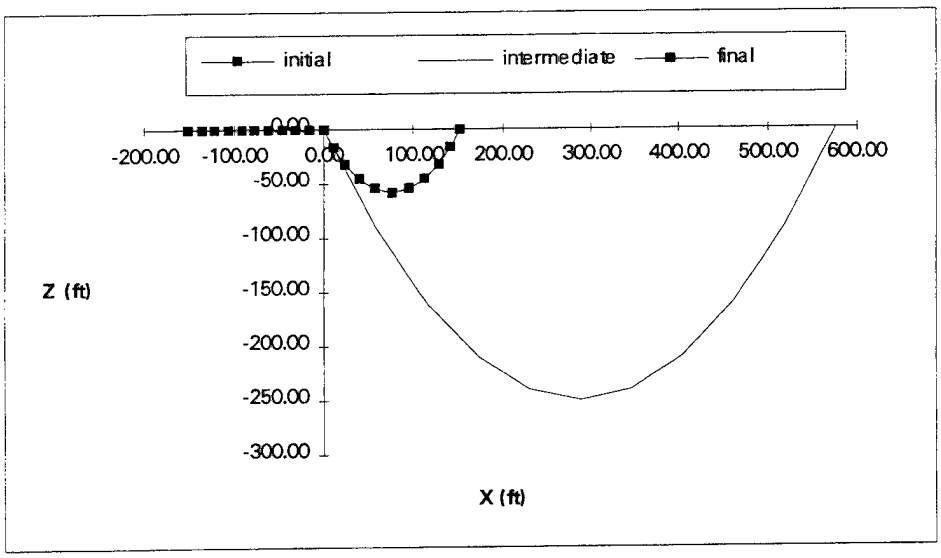


Figure 6.8 Substep solutions without event control -- varying-span cable.

6.2.2 Ill-Conditioned Solution Paths. Next we demonstrate that we can maintain solution stability for even a severely ill-conditioned solution path. We start this test case with the final solution of the previous test case. We fix the moving support, then reverse the direction of gravity. We reverse gravity by adding a distributed force in the opposite direction of gravity with a magnitude that is twice the unit weight of the cable. Given this loading, the cable catenary goes from hanging downward to "hanging" upward as shown in Figure 6.9. Each element of the cable "floats" through a "zero" tension at some point during the iterative nonlinear solution. Most traditional nonlinear solution methods diverge in attempting to solve this problem. Using the new finite cable element and its associated nonlinear solution strategy with event control, we obtain the correct solution with one static load step containing only 15 iterative substeps.

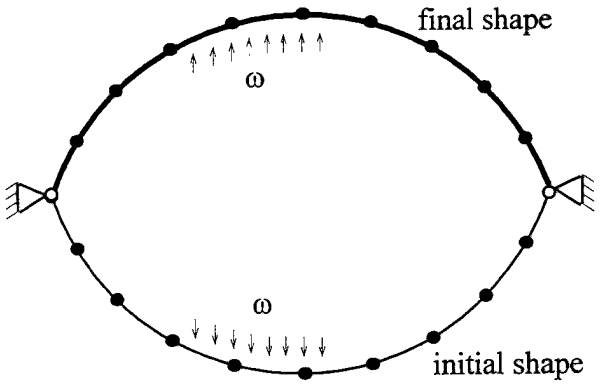


Figure 6.9 "Pop-through" of catenary cable.

6.3 Cable Whip

The hanging cable is an excellent test problem for studying low tension cable behavior. This problem uses a flexible cable hanging under its own weight. Cable tension is equal to the weight of the cable at the top and zero at the bottom. With this test problem, we show solution of a simple problem that defies solution by traditional methods.

Previous research work demonstrated that it was necessary to include pseudo-bending stiffness into the hanging cable in order to dominate the form of the solution near zero tension points (Triantafyllou and Triantafyllou, 1991). Using small amplitude perturbation analysis and pseudo-bending stiffness, it is possible to obtain results with only very small displacements and rotations (Triantafyllou and Howell, 1992).

One can argue that there is bending-type resistance in cables, but its effect is generally local and rarely significant to global response. "Localized" means that bending-type resistance occurs over very small lengths of cable, lengths less than the diameter of the cable. Since a cable consists of several yarns (or fibers) twisted together, the cross-section of a cable does not remain plane with curvature of the cable as required for traditional Euler bending theory. Instead, inelastic fiber-on-fiber shear interaction (Liu, 1995) is the local mode of resistance in a cable, not Euler bending.

We formulate our own hanging cable problem, one that demonstrates very large displacements and rotations. Figure 6.10(a) depicts our test problem. In an attempt to model whipping behavior associated with extreme cable curvature, we discretize the cable into 30 finite cable elements.

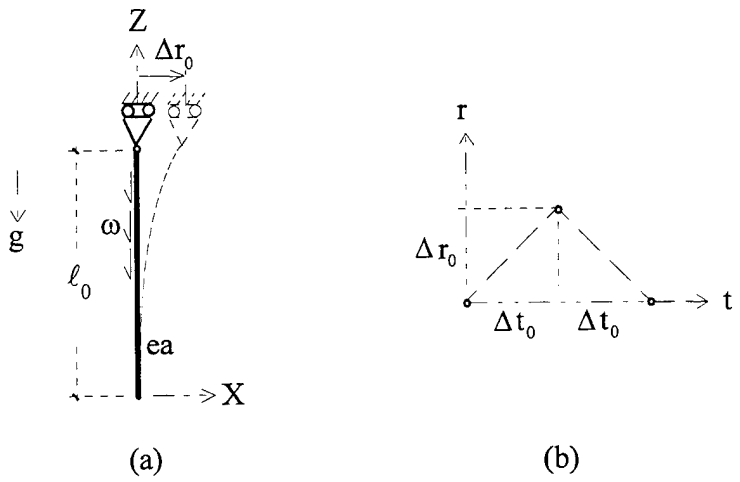


Figure 6.10 Test problem -- cable whip.

Where:

$ea = 1.131 \times 10^7$ lb = stiffness factor

$\omega = 1$ lb / ft = weight per unit length

$\ell_0 = 250$ ft = unstretched cable length

$g = 32.2$ ft / s^2 = acceleration due to gravity

$\Delta t_0 = 2s$ = first two time steps

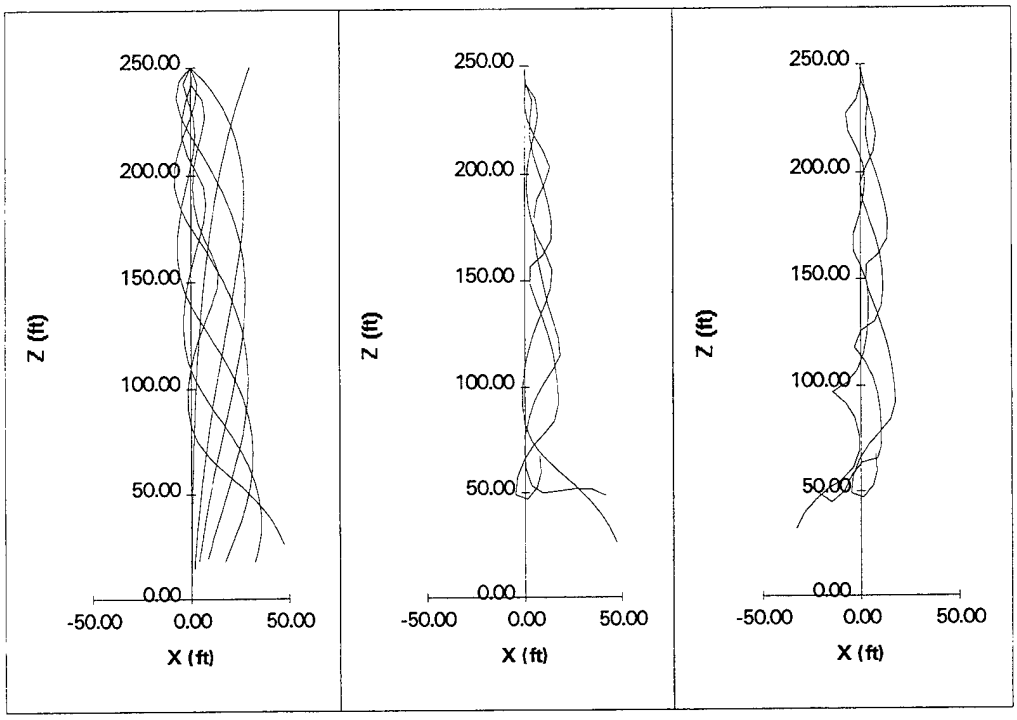
$\Delta t = 1s$ = subsequent time steps

$\Delta r_0 = 30$ft = initial impulsive movement

Given one strong initial movement at the upper end of a cable, what is the motion of the cable over time? We form the initial movement at the upper end of a cable using two sudden back-and-forth lateral displacements. Each displacement has a magnitude of 30 feet and a duration of two seconds as shown in Figure 6.10(b).

For dramatic presentation of the resulting cable whipping motion, we choose a time step size equal to one second thereafter. We can arbitrarily choose this large step size because solution stability is generally not dependent on step size. For more accurate results, we recommend a smaller time step size, one more physically consistent with the rapid whipping motion of the hanging cable.

Figure 6.11 displays progressive snapshots of the motion at each simulated time step. In this figure, the first set of snapshots (a) shows the displacement bulge moving down to the free end of the cable. The second set of snapshots (b) shows how the end of the cable rolls up just prior to the whip snapping. The third set of snapshots (c) shows the end of the cable snapping through to the other side. With no damping in this test model, the cable will continue to whip back and forth for all time.



(a) (b) (c)

Figure 6.11 Cable shape over time -- cable whip.

6.4 Suspended Cable

A suspended cable is an excellent test problem for studying the local/global computational framework. The problem is to find the static shape for a single, suspended, and weightless cable under different load cases. We can choose to represent the single cable structure by one single cable model or by two serially-connected cable submodels as shown in Figure 6.12. We can also choose to represent each cable submodel as either an analytical catenary super-element or as a set of serially-connected finite cable elements (i.e., numerical cable super-element). With this test problem, we show excellent solution accuracy, no matter which load case, or which modeling choice we make.

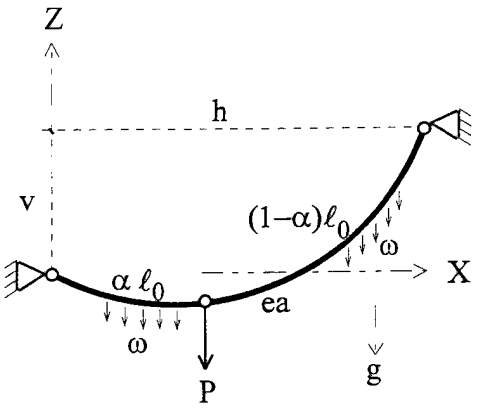


Figure 6.12 Test problem -- suspended cable.

Where:

$\omega = 0.5$ lb / ft = distributed load

P = 50 lb = point load

ea = 300 lb = stiffness factor

$\ell_0 = 100$ ft = cable length

h = 60 ft = horizontal span

v = 60 ft = elevation change

$\alpha = 0.5$ = substructure factor

Figure 6.13 shows the static shape that results from three different load cases:

- Point load
- Distributed load
- Point plus distributed load

Some modeling choices require that the solution begin with a well-posed starting configuration, hence the initial shape shown in Figure 6.13.

To get each cable shape, we use each of the following computer codes/models:

- SEASTAR or SEADYN with 20 traditional truss finite elements
- DRAIN-2DX with one or two new cable catenary super-elements
- MBDSIM with one or two submodels of 10 finite cable elements each

Solution accuracy is independent of the choice of analysis model. Instead, the chosen force equilibrium tolerance (0.001 pound) sets the final accuracy of all solutions. We choose the following goal for acceptable nonlinear solution efficiency:

- One iteration (substep) per degree of freedom

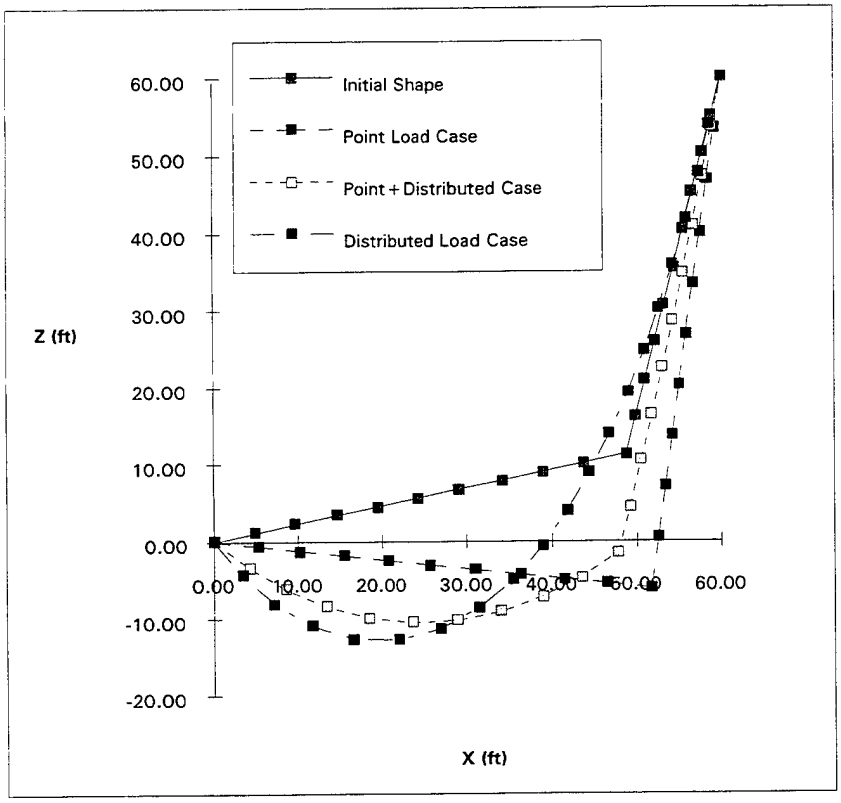


Figure 6.13 Cable shapes with different load cases -- suspended cable.

Ten elements per cable submodel is adequate for accurately representing this cable catenary. The maximum acceptable number of iterations for this test problem is thus 42. Iteration counts in excess of this number indicate an unacceptable nonlinear solution strategy. Since the exact number of iterations depends on the exact choice of solution parameters, any nonlinear strategy that generally meets this goal is acceptable.

6.4.1 Using Traditional Finite Element Models. Using 20 traditional truss-type finite elements in the computer code SEASTAR, we analyze the test problem for each of the three load cases. Because of similar modeling theories, SEADYN generally produces the same computational results as SEASTAR. To achieve our iteration goal, we must apply the load cases in a particular order to pre-condition the model for each load case. We must also start with a well-posed starting configuration, specifically the initial shape shown in Figure 6.13 and a spatially constant pretension of 50 pounds.

First we apply the point load case. Given the resulting solution for the point load case, we then add the distributed load. Given the resulting solution for the point-plus-distributed load case, we then remove the point load. The final solution is for the distributed load case only.

For this simple cable problem, it is easy to determine a suitable order of applying the various load patterns to stablize the solution. However, for a very large three-dimensional compliant structure, this process becomes unacceptably difficult and requires a structural analyst with great familiarity with the type of structure being analyzed.

6.4.2 Using New Semi-Analytical Super-Element. Using the new cable catenary super-element within DRAIN-2DX, we re-analyze the test problem for each of the three load cases. This model has a reduced set of unknowns, namely the tension and position at only the end nodes of the super-elements. We specify a reasonable starting position for only these end nodes.

In the distributed load case, we can use one cable catenary super-element. In the other two load cases, we must use two cable catenary super-elements, one above and one below the point load. With two super-elements, the distributed load is internal to each cable super-element and the point load is an external or global load.

For the point load case, numerical instability causes lack of solution convergence. The reason for this instability is that semi-analytical catenary expressions are natural for representing slack cables but unnatural for taut cables. The semi-analytical catenary super-element has limited applicability to loading cases more complex than the very simple ones presented in this test problem.

6.4.3 Using New Numerical Super-Element. Using the new cable submodel in the computer code MBDSIM, we re-analyze the test problem for each of the three load cases. Unlike the semi-analytical super-element, we can choose to use one or two cable submodels to model the problem for all load cases.

Using one submodel, there is no global solution and all 20 finite cable elements become part of the one cable submodel and its local solution. All loads are internal to this cable submodel. Using two submodels, we include only 10 finite cable elements in the local solution of each cable submodel. With two sub-models, the distributed load is internal to each cable submodel and the point load is an external or global load.

Correct solutions for all load cases require less than our goal of 42 iterations. The number of iterations is rather unrelated to choice of starting configuration and cable model. Requiring a local/global solution, the two-cable submodel needs twice as many iterations as the one cable submodel. We implemented only a rudimentary local/global solution strategy. Future research work on local/global solution strategies could improve this computational performance.

6.5 Plowing The Seafloor

Towing a plow along the seafloor is an excellent test problem for studying the interaction of several different kinds of substructures. In this problem, a cable substructure connects a seafloor plow substructure to a tow-ship substructure as shown in Figure 6.14. Each substructure has its own unique loading and resistance forces. Complex dynamic behavior at both the local

and global domains plays a large role in determining how efficient any given design or operating procedure is for plowing the seafloor.

Plowing the seafloor is useful for burying a marine cable or for collecting seafloor minerals. With this test problem, we show excellent solution robustness in the global solution even for strong local nonlinear behavior.

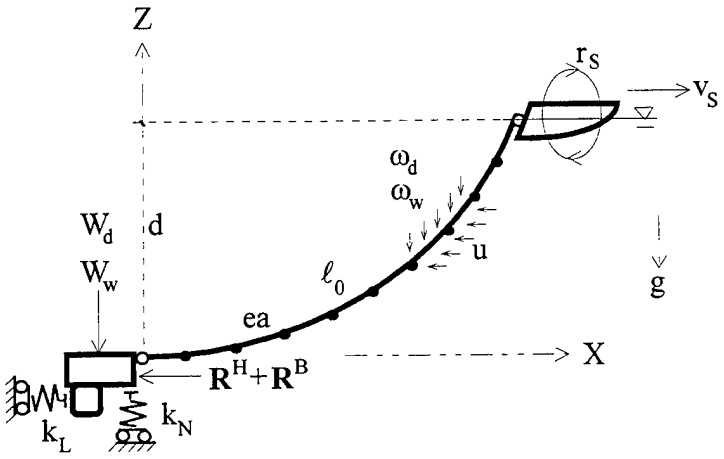


Figure 6.14 Test problem -- towing a seafloor plow.

Where:

$d = 100$ ft = water depth

$ea = 1.131 \times 10^7$ lb = cable stiffness factor

$\ell_0 = 250$ ft = unstretched cable length

$k_L = 90{,}478$ lb / ft = soil stiffness longitudinal to seafloor

$\mu = 0.3$ = Coulomb friction factor for k_L (see Section 3.6.1)

$k_N = 904{,}780$ lb / ft = soil stiffness normal to seafloor

$g = 32.2$ ft / s^2 = acceleration due to gravity

$\omega_d = 2$ lb / ft = dry weight of cable per unit length

$\omega_w = 1$ lb / ft = wet weight of cable per unit length

$W_d = 12{,}500$ lb = dry weight of plow

$W_w = 10{,}000$ lb = wet weight of plow

$\mathbf{R}^B$ = seafloor plow blade force (see Section 3.6.1)

$\mathbf{R}^H$ = seafloor plow hydrodynamic force (see Section 3.6.1)

For this test problem, we use the following specific values for tow-ship velocity, current velocity, and imposed tow-ship motion:

$r_s = 0$ = imposed motion of tow- ship

$u = 0$ = velocity of current

$v_s = 3$ ft / s = velocity of tow- ship

For the results given in Section 6.5.3, the tow-ship and seafloor nodes are fixed and thus their velocities are equal to zero. We compute hydrodynamic loading using standard force coefficients for cables (see Morison equation, Section 2.2.2.3). We compute the seafloor plow force as described in Section 3.6.1.

6.5.1 Pulling Plow out of Seafloor. Figure 6.15 shows plow velocity versus displacement in the direction of tow. This figure shows the jerking motion that results when the tow-ship, initially at rest, begins moving, suddenly tightening the cable, and jerks the plow from its resting position on the seafloor. With decreased frictional drag, the plow "jumps" forward. Then the cable slackens, and the plow's forward motion drops dramatically, as the plow slowly sinks back into the seafloor. The cable then re-tightens, once again jerking the plow forward. This jerking motion repeats itself very regularly and indefinitely. To capture the jerking motion, we choose a time step size of 0.002 second for the step-by-step simulation.

The magnitude of this jerking motion is very sensitive to the modeling parameters assigned to frictional drag, viscous drag, seafloor stiffness, cable elasticity, and Rayleigh-type damping. For the response shown in Figure 6.15, we use a stiffness proportional damping coefficient of 0.05 and no mass proportional damping coefficient. Bouncing along the seafloor is not the desired system performance if one wants to efficiently plow the seafloor. However, this result dramatically shows the kind of severe jerking nonlinear behavior that the model can reliably simulate.

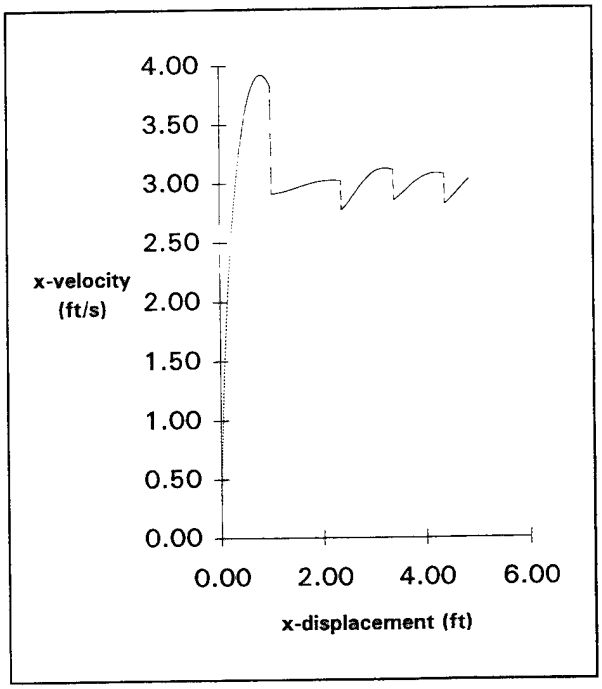


Figure 6.15 Jerking frictional behavior -- towing a seafloor plow.

6.5.2 Towing Plow Over A Seafloor Bump. Figure 6.16 shows the motion of the plow as we tow it over a bump on the seafloor. For this response simulation, we use a set of coefficients for Rayleigh type damping that produce a smooth motion of the seafloor plow. Specifically, we use a stiffness proportional damping coefficient of 0.005 and a mass proportional damping coefficient of 2.0. The jerking motion has not gone away; rather its effect has disappeared into the thickness of the lines used for plotting the results.

For more realistic tow-ship movement, we increase the tow-ship speed from zero to full speed over a one-second time interval. To capture proper seafloor elasticity effects, we choose a time step size of 0.01 second for the step-by-step simulation.

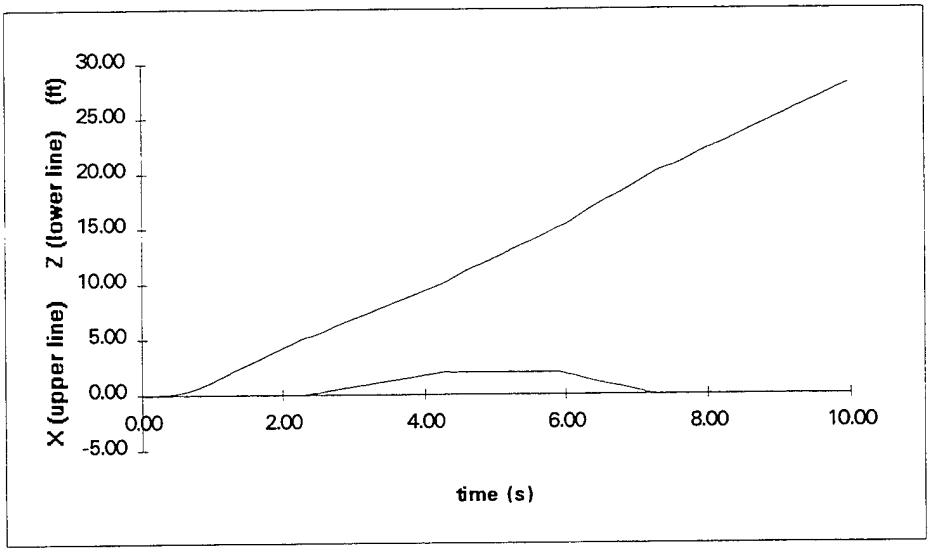


Figure 6.16 Plow displacement over a bump -- towing a seafloor plow.

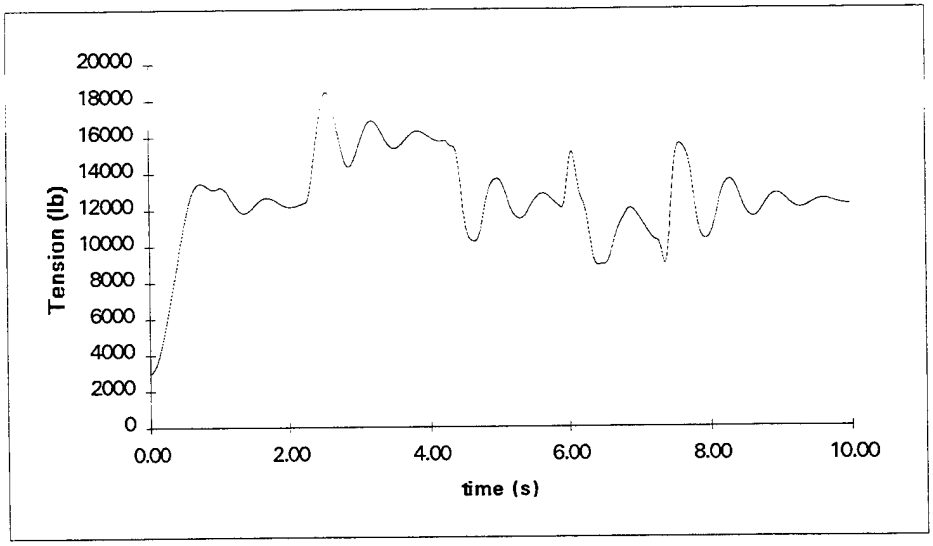


Figure 6.17 Cable tension over a bump -- towing a seafloor plow.

Figure 6.17 shows the resulting tension at the bottom of the tow-cable. The tension is higher as the plow climbs the bump and lower as it descends the bump. Abrupt changes in the seafloor slope cause the obvious tension spikes in the tow-cable that take some time to damp away. The tension time history shows that these discrete changes in seafloor slope can have quite a dramatic effect on cable tension.

On a personal computer (33-Hertz, Intel-486 chip running MS-FORTRAN compiled code), this simulation executes faster than real-time. In other words, for each minute of computer time, MBDSIM produces more than one minute of simulation output. Compared to equivalent executions with most other computer analysis programs, this is easily in the order of 10 to 1,000 times faster. Of-course, this comparison assumes equivalent complexity and discretization in the physical model. A ship-board tool for directing ship maneuvers during cable paying operations is now possible with the real-time simulation capability provided here.

6.5.3 Cable Paying. Cable paying involves adding (pay-out) or removing (pay-in) segments of cable at the tow-ship node. Since the results are much less dramatic if we allow the seafloor plow to move in a compliant fashion, we choose to fix the motion of the seafloor plow. The standard Morison equation (as given in Section 2.2.2.3) provides most of the damping. For more damping, we set the stiffness proportional damping coefficient to 0.02 and the mass proportional damping coefficient to 0.2. To capture lateral vibration of the final taut cable, we choose a time step size of 0.1 second for the step-by-step simulation.

For demonstration of cable pay-in, we make the cable slack by first paying-out two feet of cable in a static solution step. This gives the 250-foot cable a visible static catenary shape as shown in Figure 6.18. We then pay cable into the tow-ship at ten ft/s in 0.1 second dynamic time steps. Figure 6.18 shows snapshots of the resulting cable vibration at each time step. The resulting motion is similar to what happens when we rapidly tighten a very loose guitar string. For this particular result, we have set the drag coefficient to zero in the Morison equation to reduce the drag force normal to the cable and allow for rapid tightening of the cable. This rapid tightening produces a more interesting plot.

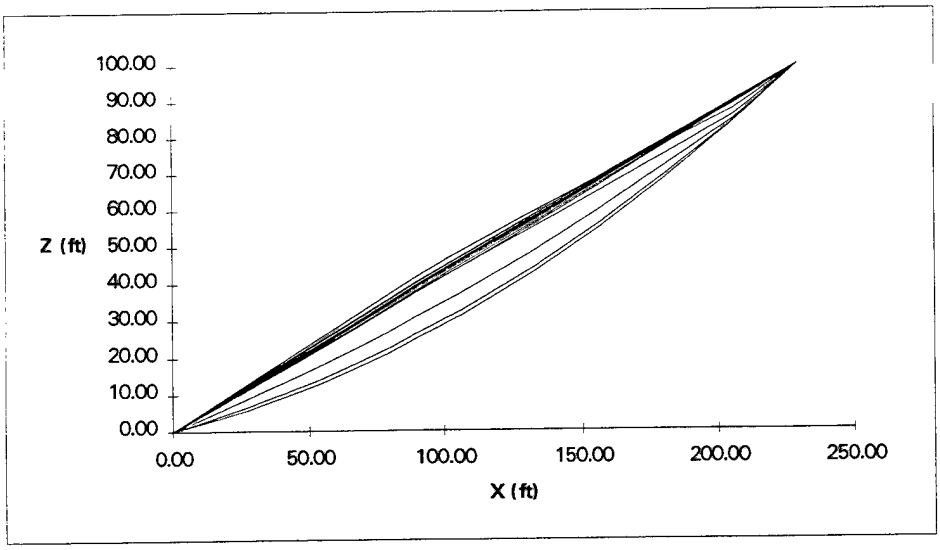


Figure 6.18 Cable profile due to pay-in of cable.

For demonstration of cable pay-out, we dynamically pay cable out from the tow-ship. With MBDSIM, we can choose an almost limitless combination of cable paying rate and time step; all produce stable realistic results. In fact, we can easily pay-out cable faster than the hydrodynamic resistance will allow it to fall towards the seafloor. The cable just piles up at the pay-out point until enough time passes to allow it to fall through the water as shown in Figure 6.19. In this figure, we pay cable out from the tow-ship at the rate of ten ft/s. The figure shows snapshots of the resulting cable profile at each 0.1 second time step. For this particular result, we use all normal coefficients for the Morison equation (as given in Section 2.2.2.3).

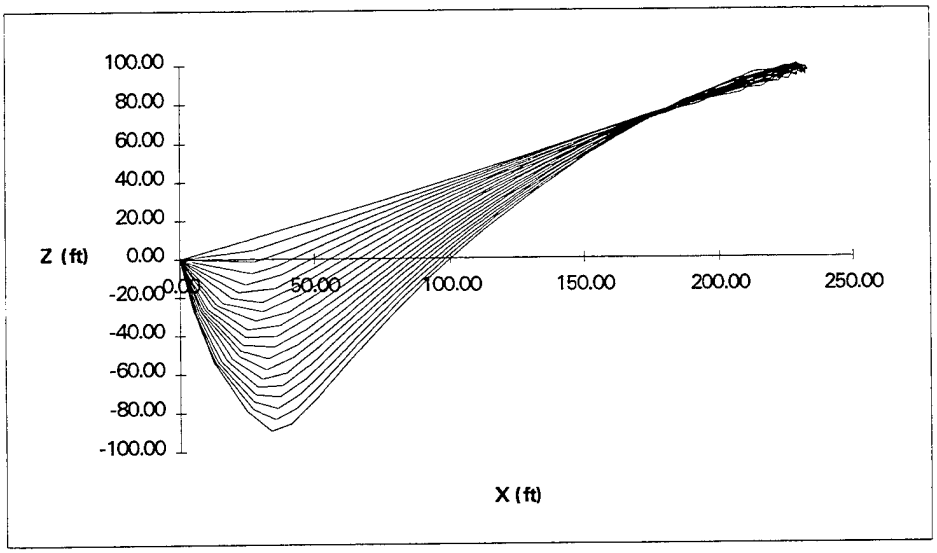


Figure 6.19 Cable profile due to pay-out of cable.

Since the drag force tangential to the cable is much smaller than drag force normal to the cable, the paid-out cable slides faster in the tangential direction than in the normal direction. To dramatize this effect, we allow the cable to fall below the seafloor. At the end of the 50-second simulation, we have paid out 500 feet of new cable (twice the original length of cable) and created 10 new elements representing the paid-out cable. Due to the large amount of hydrodynamic drag, most of these new elements remain in a slack state piled up near the pay-out point.

7. CONCLUSIONS

This report provides three key products:

- a robust nonlinear simulation strategy
- a cable simulation black-box
- a local/global computational framework

7.1 Robust Nonlinear Simulation Strategy

Newton's laws provide for stable and deterministic behavior of all structures. However, the same assurance of stability and determinism does not necessarily apply to the mathematical models of these structures. Nevertheless, given the computational methodology presented in this report, we can now expect robust (i.e., fast, efficient, and stable) simulation of compliant marine structures.

We use partial differential equations to represent full nonlinear structural behavior. We use the finite element method and the trapezoidal rule to discretize the continuous partial differential equations in space and time, respectively. This creates a step linear system of progressively changing algebraic equations, that we can incrementally solve to obtain a nonlinear step-by step simulation of structural motion.

Allowing for severe nonlinearities, we may create severe discontinuities in the mathematical model that do not exist in the true physics of the problem. Some of these discontinuities are inherent in the basis of the approximate numerical techniques used. For example, what happens to a finite element when its spatial dimensions approach zero? Other discontinuities may result from poor modeling choices as compared to the true physics of the structure. For example, what happens to the stiffness of a truss finite element (traditionally used to model cables) when the cable goes slack?

A high level of physical complexity in the mathematical model is essential for minimizing discontinuities. This often requires expanding the mathematical model to include additional physical behavior that removes the discontinuity. For example, the gravity load field creates tensile force in even the slackest or smallest of cables. No matter how small, this tensile force is essential for naturally conditioning the physical response of cables. Unlike traditional finite elements, our new finite cable element uses gravity load to condition the element stiffness.

Although we desire physical complexity in the mathematical model, we desire simplicity in the strategy for its numerical solution. Simplicity in the numerical solution is essential for maintaining a direct relationship between numerical and physical stability. As one of the simplest solution strategies, the classical Newton-Raphson method is not very robust for extremely nonlinear structural systems. By adding event control to the Newton-Raphson method, we improve its robustness in a revolutionary way.

Events are a signal to update the system of algebraic equations. By updating the equations at just the right sub-step, the simulation closely follows the true response path and thus maintains solution stability. Element events can represent any finite change from one element state to another. We recommend a natural basis for all element events. Providing a necessary connection between the mathematical model and its nonlinear solution, natural element events help maintain a level of numerical stability that approaches the physical stability of the structure itself.

For example, we use a natural "slack/taut" definition of length for our new finite cable element. Given any displacement of the element, the distance between element nodes as compared to its slack/taut length determines whether the element is in a taut or a slack state. Since the difference in stiffness between a taut and a slack cable can be very large, slack/taut element events are essential for robust nonlinear simulation of cables.

Events must recognize different scales of time and space. What appears to be a continuous stiffness change at a micro-scale may appear as an obvious discontinuity at a macro-

scale. When properly implemented, a Newton-Raphson strategy with event control ensures robust nonlinear step-by-step solution of either the static or dynamic equations of motion. There is no need for special procedures such as cutting the size of the time step.

7.2 A Cable Simulation Black-Box

The new finite cable element with its natural element events is excellent for formulating a reliable structural submodel of a cable substructure. The improved Newton-Raphson method with event control is a very effective nonlinear solution strategy for this cable submodel. Combined, the two innovations create a reliable cable simulation black box. Given any set of loads (object data), this object-oriented black box can effectively generate a realistic structural response (object action).

The cable black box requires no specific estimate of the initial shape, pre-tension or pre-stiffness. Any starting configuration, within obvious geometrical and tension realism, is permissible. The cable black box solves classical cable problems with only a small fraction of the iterations required by traditional methods. Given any state and any step increment, the number of substep iterations is usually less than the number of degrees of freedom in the submodel. The black box can even solve difficult cable problems that previously defied solution.

For the cable black box, nonlinear solution stability is generally unconditional of time step size. In other words, time step size may be as large as desired to speed up the simulation. However, time step size must be small enough to capture the periodic nature of all desired physical responses. For example, we need a time step of about 0.01 second to observe 0.1-second vibrations. A larger step size would incorrectly make high frequency vibrations appear as low frequency vibrations, i.e., alias the motion.

Since solution stability is unconditional of time step size, it is now finally possible to create a real-time simulation tool for directing ship maneuvers during oceanic cable laying operations. To lay the cable in a desired pattern on the seafloor, we must be able to predict cable motion rapidly enough to prescribe the correct course for the ship. With our new cable submodel, we can now simulate the required cable response at a rate faster than real time and at a physical modeling level greater than required.

Theoretically, we can add a cable modeling capability to any existing structural analysis program simply by adding the cable black box. However, the base program needs a local/global framework to ensure robust computational performance both inside and outside the cable black box.

7.3 The Local/Global Computational Framework

For general nonlinear simulation of compliant structures, we recommend a local/global computational framework that concentrates local behavior of substructures into submodels. A local/global computational framework allows for better nonlinear physical modeling, reduced computational expense, and improved solution stability as compared to a traditional single domain approach. Specialized for each substructure, each submodel can represent a different local structural behavior.

Extreme nonlinearities in a particular substructure often require a special strategy (e.g., very small time step) for accurate and stable simulation of the entire structure. With a

local/global computational framework, each submodel deals privately with the specific peculiarities of its local nonlinear solution. As such, we do not burden the overall global simulation with the extreme local nonlinear solution requirements of one submodel. By cycling between the local submodel and the global model, we can reliably and efficiently obtain stable structural simulations.

At the global solution level, a super-element represents a condensed version of the local submodel. The condensation procedure is simple for a static super-element and more involved for a dynamic super-element. A major computational advantage of a local/global approach is that only those submodels that are currently changing substantially from step-to-step need to be active.

Compliant marine structures lend themselves particularly well to a local/global modeling approach. In particular, compliant ocean structures often consist of diverse substructures that require diverse local submodels, each with a unique computational need. In addition, cable substructures provide the loosely coupled connections between other marine substructures that allow for efficient local/global interaction.

7.4 Other Applications

The nonlinear solution strategy and the local/global computational framework described in this report are applicable to many other types of structures. In particular, the following structural investigations could benefit greatly from our research results:

Analysis of contact forces between marine vessels and/or piers
Transient simulation of automobile crashes
Simulation of the unfolding of space station structures
Capacity analysis for re-qualification of offshore jackets
Fiber modeling for concrete bridge capacity determination

We use cables to moor marine vessels to each other or to a pier. As the vessel reacts to environmental forces, these mooring cables will suddenly go from taut to slack and vice versa. The new cable finite element would be perfect for this application. In addition, the vessel can make compressive contact with another vessel or the pier. Because of the extreme change in stiffness resulting from this contact, traditional solutions of this contact problem often fail to converge. By representing each contact as a discrete element event and using event control, we should get a very stable nonlinear solution.

An event-controlled nonlinear solution strategy is perfect for any kind of structural problem dominated by discrete contacts. For example, transient simulation of an automobile crash involves many such contacts between large-displacement shell elements. The discrete change in stiffness for each of these contacts is an obvious element event.

The truss-type structure for an orbiting space station represents another obvious application for our research results. The structural analysis of the unfolding of these highly compliant structures produces nonlinear computational difficulties similar to those that have traditionally plagued compliant marine structural analysis. These common difficulties include large-scale rotations and rapid changes in stiffness. Consequently, the same local/global

innovations developed for compliant marine structures should work for unfolding simulation of truss-type space structures.

A local/global computational framework offers excellent benefits for nonlinear static push-over analysis of offshore jacket structures. A local submodel would be excellent for representing the local buckling and re-buckling behavior of each tubular strut of a jacket structure. Only those submodels for strut substructures that are near their buckling capacity would be active.

Ultimate capacity of reinforced concrete depends on the yielding behavior of the steel rebar and its local interaction with the confining concrete. This local interaction requires concrete and steel fiber elements for locally modeling the composite nature of a member cross-section. A local/global computational framework would be excellent for representing these local submodels as super-elements in the global model of a concrete bridge or building.

In using our research results, one must remember two important modeling points. The local nonlinear solution strategy requires specific event information from the element itself that most finite element formulations do not currently provide. In addition, the element must address each discontinuity that may arise from the mathematical choice of using a discrete numerical model to solve an otherwise continuous differential problem.

7.5 Future Implications

A local/global computational framework allows for optimal choice of modeling theory and solution strategy for all kinds of local domains. However, the nonlinear solution strategy is only as robust as the local engineering submodels that represent each local structural effect.

For robust simulation of compliant structures, we need to formulate the local submodels in a physically complex and numerically simple basis. Many popular engineering theories do not possess the required modeling complexity to represent the local physical phenomenon adequately for stable step-by-step simulation.

For example, the Morison equation is a rather inadequate representation for true hydrodynamic load (and resistance) of a moving, rotating, and flexible finite element. Similarly, the Coulomb friction model is a rather inadequate representation for time-varying frictional resistance.

A local/global interface provides for easy object-oriented organization of structural analysis computer code. Using this local/global interface, developers of structural analysis software can reliably develop very complex structural models. They can independently develop specialized submodels and solution strategies as local objects without worrying about how these local submodels integrate into the global model.

Multi-processor computers offer tremendous computational advantage over single-processor computers. Much of the current research in parallel processing focuses on purely numerical methods for dividing the computational task into parallel sub-tasks assigned to each processor. Traditional algorithms for multi-processor computers seek to divide the computational task and to ensure minimal communication between processors.

A local/global computational framework would allow for automatic subdivision of the computational tasks on multi-processor computers. Solution of each local submodel would naturally be an independent task for each processor. In addition, if submodels represent loosely connected substructures, communication between local submodels would naturally be minimal.

The application of an object-oriented paradigm to domain-specific design of software provides the greatest promise for future development of software applications on parallel processing computers.

8. REFERENCES

Argyris J. and Mlejnek H.P. *Dynamics of Structures*. New York: North-Holland, 1991.

Argyris J. and Scharpf D. "Large Deflection Analysis of Prestressed Networks." *Journal of the Structural Division*, ASCE. V98 NST3, March, 1972.

ASCE, "Cable-Suspended Roof Construction, State-of-the-Art." *Journal of Structural Division*, ASCE. V97 NST6, Paper 8190: 1715-1761, June 1971.

Bathe K. *Finite Element Procedures in Engineering Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

Bathe K. and Wilson E. *Numerical Methods in Finite Element Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

Belytschko T. "An Overview of Semidiscretization and Time Integration Procedures." *Computational Methods for Transient Analysis*. Eds. T. Belytschko and T.J.R. Hughes. New York: North Holland, 1983.

Belytschko T. "A Review of Recent Developments in Time Integration." *State of the Art Surveys on Computational Mechanics*. Eds. A.K. Noor and J.T. Oden. New York: American Society of Mechanical Engineers, 1989.

Belytschko T. and Hughes T.J.R., personal communication in conjunction with a short course entitled, "Nonlinear Finite Element Analysis." Palo Alto, CA, 26-29 July, 1993.

Borgman L., Shields D., Bartel W. and Zueck R.F. "The SIMBAT Software Package for Stochastic Interpolation of Ocean Wave Kinematics as an Aid in the Engineering Design of Large Floating Structures." *Engineering in the Oceans V*, ASCE. College Station, Texas, 1992.

Carrera E. "A Study of Arc-Length-Type Methods and their Operation Failures Illustrated by a Simple Model." *Computers and Structures*. V50 N2: 217-229, 1994.

Chakrabarti S.K. *Hydrodynamics of Offshore Sructures*. Berlin: Springer-Verlag, 1987.

Chakrabarti S.K. *Nonlinear Methods in Offshore Engineering*. Amsterdam; New York: Elsevier, 1990.

Chiou R. "Nonlinear Hydrodynamic Response of Curved Singly-Connected Cables." Ph.D. diss., Oregon State University, No. OE-90-01, 7 Dec. 1989.

Clough R.W. and Penzien J. *Dynamics of Sructures.* New York: McGraw-Hill, 1975.

Cook R.D., Malkus D.S., and Plesha M.E. *Concepts and Applications of Finite Element Analysis*. Chichester, England; New York: Wiley, 1989.

Craig R.R. "Substructure Methods in Vibration." *Journal of Design Engineering Division, Special 50th Anniversary Design Issue*, ASME. V117: 207-213, June 1995.

Crisfield M. *Non-linear Finite Element Analysis of Solids and Structures*. Chichester, England: Wiley, 1991.

Crisfield M. and Shi J. "Co-Rotational Element/Time-Integration Strategy for Non-Linear Dynamics." *International Journal for Numerical Methods in Engineering.* V37: 1897-1913, 1994.

Dahlquist G. "A Special Stability Problem for Linear Multi-Step Methods." *BIT*, 3: 27-43, 1963.

De Roeck Y., Le Tallec P. and Vidrascu M. "A Domain-Decomposed Solver for Nonlinear Elasticity." *Computer Methods in Applied Mechanics and Engineering.* V99 N2-3, Sept. 1992.

Dickens J. "Numerical Methods for Dynamic Substructure Analysis." Ph.D. diss., University of California, Berkeley, CA, 1980.

Duff I.S., Erisman A.M. and Reid J.K. *Direct Methods for Sparse Matrices*. Oxford; New York: Clarendon, 1989.

Farhat C. and Roux F. "A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm." *International Journal for Numerical Methods in Engineering.* V32: 1205, Oct. 1991.

Fenves G. "Object-Oriented Programming for Engineering Software Development." *Engineering With Computers*. V6 N1: 1-15, 1990.

Friedberg S.H., Insel A.J. and Spence L.E. *Linear Algebra.* 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1989.

Fuchs G., Roy J. and Schrem E. "Hypermatrix Solution of Large Sets of Symmetric Positive Definite Linear Equations." *Computer Methods in Applied Mechanics and Engineering.* V99 N2-3: 197-206, Sept. 1972.

Garrison C.J. "Hydrodynamic Loading of Large Offshore Structures -- Three Dimensional Source Distribution Methods." *Numerical Methods in Offshore Engineering.* Eds. R. Zeinkiewicz, R. Lewis and K. Stagg. Chichester; New York: Wiley, 1978.

Guyan R.J. "Reduction of Stiffness and Mass." *American Institute of Aerospace and Astronautics Journal.* V3 N2: 380, Feb. 1965.

Hajjar J. and Abel J. "On the Accuracy of Some Domain-by-Domain Algorithms for Parallel Processing of Transient Structural Dynamics." *International Journal for Numerical Methods in Engineering.* V28 N8: 1855-1874, Aug. 1989.

Hudspeth R.T. and Shields D.R. "A Critical Review of the Relative Motion Morison Equation." *Ocean Structural Dynamics Symposium.* Oregon State University, Corvallis, OR, Sept. 1986.

Hughes T.J.R. "Analysis of Transient Algorithms with Particular Reference to Stability Behavior." *Computational Methods for Transient Analysis.* Eds. T. Belytschko and T.J.R. Hughes. New York: North Holland, 1983.

Hughes T.J.R. and Belytschko T. "A Precis of Developments in Computational Methods for Transient Analysis." *Journal of Applied Mechanics*, ASCE. V50/1033, Dec. 1983.

Humar J.L. *Dynamics of Structures.* Englewood Cliffs, NJ: Prentice Hall, 1990.

Irvine H.M. *Cable Structures.* Cambridge, MA: MIT Press, 1981.

Keys D.E., Chan T.F., Meurant G., Scroggs J.S and Voigt R.G., eds. *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations.* Philadelphia: Society for Industrial and Applied Mathematics, May. 1991.

Leonard J.W. *Tension Structures: Behavior and Analysis.* New York: McGraw-Hill, 1988.

Leonard J.W. and Karnoski S.R. "Simulation of Tension Controlled Cable Deployment." *Applied Ocean Research.* V12 N1: 34-42, Jan. 1990.

Leonard J.W. and Recker W. "Nonlinear Dynamics of Cables with Low Initial Tension." *Journal of Engineering Mechanics Division*, ASCE. V98 NEM2, April 1972.

Levinson D.A. and Kane T.R. "A Usable Solution of The Hanging Cable Problem." *Computers and Structures.* V46 N5: 821-844, Mar. 1993.

Liu F. *Computer Modeling of High-Strength Synthetic Lines Under Static and Cyclic Tensile Loads.* TR-2052-OCN, Naval Facilities Engineering Service Center, Port Hueneme, CA, Dec. 1995.

Morison J.R., O'Brien M.P., Johnson J.W. and Schaaf S.A. "The Force Exerted by Surface Waves on Piles." *Petroleum Transactions*. AIME, TP 2846, V189: 149-154, 1950.

Newmark N.M. "A Method of Computation for Structural Dynamics." *Journal of Engineering Mechanics Division*, ASCE, EM3, V85: 67-94, 1959.

Nour-Omid B., Parlett B.N. and Taylor R.L. "A Newton Lanczos Method for Solution of Nonlinear Finite Element Equations." *Computers and Structures*. V16 N1-4: 241-252, 1983.

Oden J.T. and Demkowicz L. "A Survey of Adaptive Finite Element Methods in Computational Mechanics." *State-of-the-Art Surveys on Computational Mechanics*. Eds. A.K. Noor and J.T. Oden. New York: American Society of Mechanical Engineers, 1989.

Oughourlian C. and Powell G.H. *ANSR-III, General Purpose Computer Program for Nonlinear Structural Analysis*. UCB/SEMM-82/21, Earthquake Engineering Research Center, University of California, Berkeley, CA, 1982.

Palo P.A. "Dynamic Cable Analysis Models." *Proceedings*. OTC 4500, Offshore Technology Conference, Houston, TX, May 1983.

Park K. and Felippa C. "Partitioned Analysis of Coupled Systems." *Computational Methods for Transient Analysis*. Eds. T. Belytschko and T.J.R. Hughes. New York: North-Holland, 1983.

Parlett B.N. *The Symmetric Eigenvalue Problem*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

Paulling J.R. "Time Domain Simulation of Semisubmersible Platform Motion with Application to the Tension Leg Platform." *Spring Meeting/Star Symposium*, pp. 303-314, Society of Naval Architects and Marine Engineers, San Francisco, CA, 1977.

Peyrot A. H. "Analysis of Flexible Transmission Lines." *Journal of the Structural Division,* ASCE. V104 NST5, May 1978.

Peyrot A.H. and Goulois A.M. "Analysis of Cable Structures." *Computers and Structures*. V10 N5: 805-813, 1979.

Peyrot A.H. "Marine Cable Structures." *Journal of the Structural Division*, ASCE. V106 NST12, Dec. 1980.

PMB Engineering, *SEASTAR Theoretical Manual*. Bechtel Corporation, San Francisco, CA, Oct. 1989.

Porter F. and Powell G.H. *Static and Dynamic Analysis of Inelastic Frame Structures.* UCB/EERC Report No. 71-3, Earthquake Engineering Research Center, University of California, Berkeley, CA, 1971.

Powell G.H. "Modeling and Solution Strategies for Nonlinear Braced Frames." *Supercomputing in Engineering Structures.* Ed. C. Brebbia. Southhampton: Computational Mechanics Publications, 1989.

Powell G.H. *Substructure Technique Development for Nonlinear Structural Analysis of Oceanic Cable Structures.* Contract Report, Naval Facilities Engineering Service Center, Port Hueneme, CA, Dec. 1995.

Prakash V. and Powell G.H. *DRAIN-2DX, DRAIN-3DX and DRAIN-BUILDING: Base Program Design Documentation.* UCB/SEMM-93/16, Earthquake Engineering Research Center, University of California, Berkeley, CA, Dec. 1993.

Prakash V., Powell G.H. and Campbell S. *Drain-3DX: Base Program Description and User Guide.* UCB/SEMM-94/07, Earthquake Engineering Research Center, University of California, Berkeley, CA, Aug. 1994.

Prakash V., Powell G.H. and Filippou F.C. *Drain-2DX: Base Program Description and User Guide.* UCB/SEMM-92/29, Earthquake Engineering Research Center, University of California, Berkeley, CA, Dec. 1992.

Prezemienieki J.S. "Matrix Structural Analysis of Substructures." *American Institute of Aerospace and Astronautics Journal.* V1: 138-147, 1963.

Rosenthal F. "The Application of Greenhill's Formula to Cable Hockling." *Journal of Applied Mechanics.* V43 N4: 681-683, 1976.

Row D.G. and Powell G.H. *A Substructure Technique For Nonlinear Static and Dynamic Analysis.* UCB/SEMM-78/15, Earthquake Engineering Research Center, University of California, Berkeley, CA, Aug. 1978.

Shugar T. "Automated Dynamic Relaxation Solution Method for Ocean Cable Structures." *Proceedings of Ocean Structural Dynamics Symposium.* Oregon State University, Corvallis, OR, 1988.

Shugar T. and Armand J.L. *Computational Methods in Ocean Structural Engineering--A Review.* TN 1768, Naval Civil Engineering Laboratory, Port Hueneme, CA, May 1987.

Simons J.W. and Powell G.H. *Solution Strategies for Statically Loaded Nonlinear Structures.* UCB/EERC 82/22, Earthquake Engineering Research Center, University of California, Berkeley, CA, Nov. 1982.

Taylor R.J. *Drag Embedded Anchor Tests in Sand and Mud.* TN-1635, Naval Civil Engineering Laboratory, Port Hueneme, CA, 1982.

Triantafyllou M.S. and Howell C.T. "Nonlinear Impulsive Motions of Low-Tension Cables." *Journal of Engineering Mechanics*, ASCE. V118 N4: 807-830, Apr. 1992.

Triantafyllou M.S. and Triantafyllou G.S. "The Paradox of the Hanging String--An Explanation Using Singular Perturbations." *Journal of Sound and Vibration.* V148 N2: 343-351, July, 1991.

Tuma, J. *Engineering Mathematics Handbook; Definitions, Theorems, Formulas, Tables.* New York: McGraw-Hill, 1970.

Underwood P. "Dynamic Relaxation." *Computational Methods for Transient Analysis.* Eds. T. Belytschko and T.J.R. Hughes. New York: North-Holland, 1983.

Webster R.L. "Nonlinear Static and Dynamic Response of Underwater Cable Structures using the Finite Element Method." *Proceedings.* OTC 2322, Offshore Technology Conference, Houston, TX, May 1975.

Webster R.L. *An Application of the Finite Element Method to the Determination of Nonlinear Static and Dynamic Responses of Under water Cable Structures.* Ph.D. diss., Cornell University, Ithaca, NY, 1976.

Webster R.L. "Analysis of Deep Sea Moor and Cable Structures." *Proceedings.* OTC 3623, Offshore Technology Conference, Houston, Texas, May 1979.

Webster R.L. "On the Static Analysis of Structures with Strong Geometric Nonlinearity." *Computers and Structures.* V11 N1/2: 137-145, 1980.

Webster R.L. *SEADYN Mathematical Models.* CR 82.019, Naval Civil Engineering Laboratory, Port Hueneme, CA, April 1982.

Webster R.L. and Palo P.A. *SEADYN90 User's Manual.* TN 1803. Naval Civil Engineering Laboratory, Port Hueneme, CA, Nov. 1989.

Wiegel R.L. *Oceanographical Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 1964.

Williams F.W. "Comparison Between Sparse Stiffness Matrix and Substructure Methods." *International Journal of Numerical Methods in Engineering.* V5: 383-394, 1973.

Wood W. "Some Transient and Coupled Problems--A State-of-the-Art Review." *Numerical Methods for Transient and Coupled Problems.* Eds. R. Lewis, E. Hinton, P. Bettess and B. Schrefler. Chichester, England; New York: Wiley, 1987.

Zueck R. F. " Local/Global Approach to Nonlinear Simulation of Compliant Marine Structures." Ph.D. diss., University of California, Berkeley, CA, 1996.

Zueck R.F. "Dynamic Response Analysis of a Small Semisubmersible Moored in Deepwater." *Ocean Structural Dynamics Symposium*, Oregon State University, Corvallis, OR, 1986.

Zueck R.F., Pawsey S. and Leverette S. "Measured and Simulated Response of a Small Semisubmersible Moored in Deep Water." *Engineering in the Oceans V*, ASCE. College Station, TX, 1992.

Zueck R.F. and Powell G.H. "Stable Numerical Solver for Cable Structures." *Proceedings of the International Symposium on Cable Dynamics*. University of Liege, Liege, Belgium, 19-21 Oct. 1995.

Zueck R.F. and Shields D. "Deepwater Semisubmersible Motion Simulation." *Oceans '84*. Marine Technology Society and IEEE Council on Oceanic Engineering, V2, Washington, D.C., 1984.

Zueck R.F. and Shields D. "Differences in Mooring Models for Deep Water Semisubmersible Motion Simulation." *Offshore and Arctic Frontiers Symposium*. New Orleans, LA: American Society of Mechanical Engineers, 1986.

Zueck R.F. and Shields D. "Analysis of Large Cable Array." *Proceedings of the International Symposium on Cable Dynamics*. University of Liege, Liege, Belgium, 19-21 Oct. 1995.